3. Negations

3.1. Not: contradicting content

3.1.0. Overview

In this chapter, we direct our attention to *negation*, the second of the logical forms we will consider.

3.1.1. Connectives

The connective we will study in this chapter is *negation*, which is associated with the English word not. As has been the case with *conjunction*, we will use the term *negation* also for the sentences produced by the operation of negation. We will represent the form of such sentences symbolically using ¬ (the *not sign*) as our sign for negation so that $\neg\,\varphi$ is the negation of φ. To indicate negations using English, we will use not as an alternative to ¬, writing it, too, in front of the negated sentence so that, in this notation, not φ is the negation of φ.

The use of the term connective for negation is standard but in some ways not very apt. The word not in English is not a combining operation; it is not a conjunction (in the grammatical sense) that serves to connect clauses but instead an adverb, a modifier of a single clause. Thus it would be a mistake to associate the term connective too closely with the ideas of connection or combination. A *connective* is better thought of as an operation that forms or generates a sentence from one or more sentences. This operation may combine or modify, and it may do both.

We will extend the terminology used for conjunction and refer, however inaptly, to any sentence generated by a connective as “compound” and refer to the one or more sentences it is generated from as “components.” When analyzing English sentences, the ultimate components we encounter may not be parts, in any grammatical sense, of the sentences we analyze. They will rather be the sentences whose logical forms we do not describe; that is, they are the unanalyzed residue of our analysis.

3.1.2. Contradictory propositions

We could base the truth conditions of negation directly on the observation that the word false means '*not* true' and the word true means '*not* false.' But it will be more enlightening to base it instead on some understanding of the logical relations between a negation $\neg\varphi$ (or not φ) and its component φ.

One obvious generalization about negation is that a negative sentence is incompatible with the component that is negated. For example, in the traditional children's story, even before sitting down to her taste test, Goldilocks knew that The porridge is too hot and The porridge is not too hot could not both describe the same bowl. Each excludes the other; they are mutually exclusive (in the sense defined in 1.2.6). We can explain this fact about negation if we assume that the negation $\neg\varphi$ of a sentence φ is false whenever the sentence φ is true. And that settles the part of the truth table for negation shown below.

φ	$\neg\varphi$
T	F

But it does not settle the rest. The sentences The porridge is too hot and The porridge is too cold are also mutually exclusive, but Goldilocks found two cases in which The porridge is too hot was false, one in which The porridge is too cold is true and another in which it was false. So the mutual exclusiveness of φ and $\neg\varphi$ is not enough to settle the truth value of $\neg\varphi$ when φ is false.

There is a second relation between a sentence and its negation that does settle this value. While the falsity of both The porridge is too hot and The porridge is too cold would leave open the possibility that the porridge is just right, The porridge is too hot and The porridge is not too hot allow no third case. That means the two sentences are jointly exhaustive of all possibilities (see 1.2.6 for this idea). This relation serves to settle the second row of the truth table for negation; if φ is false then $\neg\varphi$ must be true.

φ	$\neg\varphi$
T	F
F	T

A negation $\neg\varphi$ thus has a truth value that is always the opposite of the truth value of its component φ. In 1.2.6, we spoke of such sentences (that is, sentences that are both mutually exclusive and jointly exhaustive) as "contradictory." So a sentence and its negation are contradictory sentences; each contradicts the other. The negation of a sentence φ need not be the only sentence that contradicts φ, but any sentence that stands in this relation to φ will be logically equivalent to $\neg\varphi$.

Figure 3.1.2-1 shows the effect of negation on the proposition expressed; the possibilities ruled out by the sentence (A) and its negation (B) are shaded. The images of dice recall the example of Figure 2.1.2-1; if they are taken to indicate regions consisting of the possible worlds in which a certain die shows one or another number, the proposition shown in 3.1.2-1A is The number shown by the die is less than 4 and 3.1.2-1B illustrates the negation of this proposition.

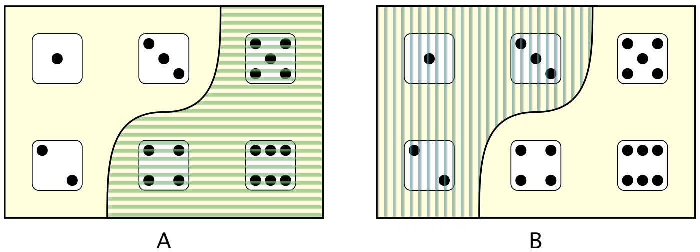


Fig. 3.1.2-1. Propositions expressed by a sentence (A) and its negation (B).

The possibilities left open by a sentence are ruled out by its negation—no possibilities are left open by both—because the two are mutually exclusive. And the possibilities ruled out by a sentence are left open by a sentence—none are ruled out by both—because the two are jointly exhaustive.

The reversal in the range of possibilities left open in moving from a sentence to its negation are the basis for what can be seen as the the key properties of negation.

Contravariance. *A negation implies the result of replacing its component with anything that component is implied by.* That is, if $\varphi \vDash \psi$, then $\neg\psi \vDash \neg\varphi$.

Involution. *To deny a negation is to assert what it negates.* That is, $\neg\neg\varphi \simeq \varphi$.

Compositionality. *Negations are equivalent if their components are equivalent.* That is, if $\varphi \simeq \varphi'$, then $\neg\varphi \simeq \neg\varphi'$.

The last of these follows from contravariance just as the compositionality of conjunction follows from covariance; and, as noted in 2.1.2, compositionality is something we would expect to hold of any connective. The distinctive character of negation is reflected in the first two principles.

In particular, contravariance and involution together tell us that $\neg\psi$ implies

$\neg\varphi$ if and only if φ implies ψ. Contravariance alone supplies the if part of this; in the other direction, the two principles tell us that, if $\neg\psi$ implies $\neg\varphi$, then φ is equivalent to something (namely, $\neg\neg\varphi$) that implies something (namely, $\neg\neg\psi$) that is equivalent to ψ. In sum, the more said by a claim, the less said by its denial; and the less said by a claim, the more said by its denial. Compare The package won't arrive next Wednesday and The package won't arrive next week. The latter is the more informative, and it denies the less informative of the two positive sentences The package will arrive next Wednesday and The package will arrive next week. Notice in the diagrams above that, as the area ruled out by a sentence increases, the area ruled out by its denial decreases, and vice versa.

Connectives that have truth tables express truth functions and are therefore said to be *truth functional*, and this is something more than being propositional. Conjunction and negation are truth functional, but not all connectives have this property. The following simple example of a non-truth-functional connective should suggest a whole range of further examples. Compare these two sentences:

The bridge is not finished
The bridge will never be finished.

The truth value of the first is determined once we know the truth value of The bridge is finished, but this is not always enough information in the case of the second. When The bridge is finished is true, we know that The bridge will never be finished is false; but, when The bridge is finished is false, we need more information to determine the truth value of The bridge will never be finished. In particular, we need at least some information about the truth value of The bridge is finished at times in the future; and before we can know that The bridge will never be finished is true, we need to know the truth value of The bridge is finished at *all* times in the future. And this means that the connective marked by the English form It will never be the case that φ is not truth functional: the actual truth value of the compound formed by it is not settled by the actual truth value of the component φ. But we would still expect the proposition expressed by it to be settled if we knew everything about the proposition expressed by its component—i.e., if we knew the truth value of its component in all possible worlds. We simply cannot limit consideration to one possibility at a time in the way we can with truth-functional connectives.

We will limit our study of connectives to those that are truth-functional. The study of such connectives is *truth-functional logic* (a phrase that was mentioned in 1.1.7). The connective expressed by It will never be the case that φ would be studied by *tense logic*, the logic of tenses and other temporal modifiers. This is one part of the logic of connectives that lies beyond truth-functional logic. Another part is the logic of modal auxiliaries like must and can. These, too, are associated with non-truth-functional connectives, and the study of the logical properties of these connectives is referred to as *modal logic*, an ancient branch of logic that became an active area of research again in the 20th century.

Glen Helman 06 Aug 2010

3.1.3. Negation in English

Many questions that arise concerning the use of conjunction to analyze English sentences do not apply to negation. In particular, since a compound formed by negation has only a single component, there is no need to worry identifying components that make independent contributions to the whole. It is important, though, to be sure that the component that we uncover is related to the whole compound in the way that negation indicates—that is, we need to make sure that the two are contradictory.

Negative prefixes on adjectives (un-, in-, a-, etc.) sometimes function as stylistic variants for not. But the effect of such a prefix may not always be to negate since the result of adding it may not always be contradictory to the original sentence. For example, happy and unhappy seem to be used sometimes as synonyms for joyful and sad. In such usage, the sentence Hal is unhappy is not the negation of Hal is happy because both might be false. The only way to distinguish such cases from ones where the prefix is a sign of negation (as in The road is unfinished) is to ask yourself whether a sentence with a negative prefix and the corresponding sentence without it jointly exhaust all possibilities in the sense that at least one of the two is bound to be true.

When doing this, it is important to remember the difference between truth and appropriateness. That is, to show that Hal is happy and Hal is unhappy are not jointly exhaustive, it is not enough to find a case where it would not be appropriate to assert either—as when Hal's state of mind is neutral—for one of the two inappropriate assertions might still be true. It would even be possible for unhappy to be appropriate in exactly the same circumstances as some term like sad even though the two had different truth conditions. While it is not easy to rule out this sort of possibility, remember that we have one test to use. Imagine being asked the two questions Is Hal happy? and Is Hal unhappy? when you know his state of mind is neutral. Ask yourself if you would reply No to both or reply No to one and Yes, but … to the other.

So the presence of a stylistic variant of not is not sufficient to indicate negation—and it is also not necessary. Some sentences can be analyzed as negations even though they do not contain either not or a stylistic variant because they contain another logical expression that introduces a negative element. For example, The road was neither smooth nor straight can be analyzed as the negation of The road was either smooth or straight. In this case, we were able to simply remove the negative element in order to identify the component to which negation is applied; but, in other cases, some restatement may be needed to formulate a component that is contradictory to the whole compound.

That is often the case when negation is introduced by way of words or phrases containing no. For instance, No one bought the book is negative, but what is it the negation of? It is not the negation of Everyone bought the book, for to deny that would be to say only that there is at least one person who failed to buy it. No one bought the book must be the negation of At least one person bought the book or, more briefly, Someone bought the book. English is regular enough on this point that you could make it a rule of thumb to treat no as indicating the negation of some, but this is not a rule to be applied without thought. Again, the best general policy is to ask yourself whether the original sentence and the component you take to be negated are really contradictories—whether it really is the case that they cannot both be true and cannot both be false.

A related problem concerns the word any. This often appears in negative sentences—such as I didn't speak to anyone. Although this sentence is a negation, it cannot be analyzed as the negation of I spoke to anyone—a sentence that is hard to understand (except in contexts where it is elliptical for something like I spoke to anyone I wanted to). Instead, I didn't speak to anyone is the negation of I spoke to someone where this is understood to mean I spoke to at least one person. The problem with retaining any in the component of a negation is that it is generally used only in the presence of certain other words—not is one, but also if and some others—and it is hard, if not impossible, to understand the force of any when it is removed from such a context. But English is fairly regular here, too; and a sentence in which any is used with not can usually be regarded as a negation whose component can be stated using some in place of any.

For this approach to no and not … any to work, it is important that some mean 'at least one'. Now, in some contexts, the fact that some is used with a singular noun can lead to an implicature of 'only one'. For example, a sentence like I spoke to someone may implicate that *only* one person was spoken to. To see that this implicature is not an implication, imagine speaking to two people and being asked, "Did you speak to someone?" I think the natural answer would be Yes rather than No—though you might add In fact, I spoke to two people if this further information was relevant. If that is right, the suggested analysis of I spoke to no one and I didn't speak to anyone does work, but the best policy is still to ask yourself whether the component you identify is really contradictory to the original sentence.

Similar issues arise when we consider the result of negating a negation (that is, the form ¬ ¬ φ or not not φ). Although we can capture some further English constructions by this form, the principle of involution in 3.1.3 tells us that we can find no new logical properties since the two forms ¬ ¬ φ and φ are logically equivalent. That is, doubling a negation cancels it. The sentence The road is not unfinished is merely a roundabout way of saying that the road is finished. It is true that double negations do not always seem to have the same force as positive statements; but this is naturally ascribed to a difference in appropriateness without a difference in truth conditions, a difference in implicatures but not implications.

To get a sense of the play of implicatures here, consider the following dialogue (with underlining used to mark emphasis):

A: Hal is not unhelpful.
B: So, in other words, he's helpful.
A: Well, yes, but he's not *really* helpful.
B: You mean he just appears to be helpful?
A: No, he's really *helpful*. He's just not *really* helpful.

This shows—if the point needed making—that truth conditions are often less the foundations of communication than walls to bounce things off. But even so, they make their presence felt—and that is what we are trying to capture. When logicians question the equivalence of a double negation and a positive statement, it is usually on different grounds.

And, surprising as it may be, the equivalence of φ and ¬ ¬ φ is actually one of the more controversial principles among logicians. A small school of mathematics called *intuitionism* grew up around efforts in the early part of the 20th century by the Dutch mathematician L. E. J. Brouwer (1881-1966) to give what he took to be a philosophically satisfactory account of the continuum (the full range of real numbers including irrational numbers like π and the square root of 2). He came to reject certain ways of proving the existence of mathematical objects, and he also rejected certain logical principles—the equivalence of φ and ¬ ¬ φ among them—which could be used to justify such proofs. Brouwer did not succeed in transforming mathematical practice or leading most logicians to doubt the equivalence of φ and ¬ ¬ φ, but his ideas have proved useful in the study of computation and have led to a deeper understanding of the significance of various logical principles concerning negation.

3.1.4. Negated conjunctions and conjoined negations

While the ability to negate a negation does not enable us to say any more—however much more we may suggest—we increase the range of propositions we can express considerably when we mix negation and conjunction. The variety of English sentences whose forms we can express naturally will still be somewhat limited, and we will go on to capture others in the next two chapters. But the variety of logical relations between compounds and their components that can be expressed using conjunction and negation will be as great as any we will see when we are considering connectives alone (that is, until chapter 6). Indeed, any connective that is truth-functional—i.e., any whose meaning can be captured in a truth table—can be expressed using conjunction and negation alone.

The real key to the power of expression of these two connectives lies in the ability to negate conjunctions, so let us look more closely at such forms. We will begin with the example It was not both hot and humid.

It was not both hot and humid
¬ it was both hot and humid
¬ (it was hot ∧ it was humid)

¬ (T ∧ M)
not both T and M

T: it was hot; M: it was humid

The parentheses and location of not before both record the fact that the sentence as a whole is a negation. That is, negation here has wider scope than conjunction and is thus the main connective.

We will refer to the way this sentence is related to its unanalyzed components as the *not-both form*. Our analysis together with the truth tables for negation and conjunction enable us to calculate a truth table for it. The table below follows the conventions for exhibiting the values of compounds that were introduced in 2.1.8. (That is, each of the two columns of values on the right is written under the sign for the connective whose table was the last used in calculating it.)

φ	ψ	¬	(φ ∧ ψ)
T	T	Ⓕ	T
T	F	Ⓣ	F
F	T	Ⓣ	F
F	F	Ⓣ	F

The plain roman Ts and Fs are the values for the conjunction $\varphi \wedge \psi$ in each case, and the circled values for the form as a whole come by following the table for negation and taking the opposite of the value of the conjunction in each row.

In the symbolic analysis of the not-both form, parentheses not only reflect the structure of the sentence analyzed but also make a significant difference in the proposition expressed. If we drop them and write $\neg \varphi \wedge \psi$ (i.e., both not φ and ψ), we will no longer be marking the conjunction as a component of a larger negation. The negation sign will instead apply (by default) to φ alone, and the main connective will be conjunction. That is, we will have a conjunction whose first component is a negation. The truth table for this form is as follows:

φ	ψ	¬φ	∧ψ
T	T	F	Ⓕ
T	F	F	Ⓕ
F	T	T	Ⓣ
F	F	T	Ⓕ

In the example we began with, dropping the parentheses gives us $\neg T \wedge M$ (that is, both not T and M), which can be put into English as follows:

¬ it was hot ∧ it was humid
It wasn't hot ∧ it was humid
It wasn't hot, but it was humid

And we will refer to the general form $\neg \varphi \wedge \psi$ as the *not-but form*.

The not-but sentence above also could be expressed (though more awkwardly) as It was both not hot and humid. (If this does not seem to make sense, try reading not hot as if it was hyphenated and pause briefly after it; that is, read it as you would It was both not-hot—and humid.) A comparison of this last (awkward) expression of the not-but form with our original not-both example is revealing:

Sentence	*Analysis*
It was not both hot and humid	$\neg(T \wedge M)$ ***or*** not both T and M
It was both not hot and humid	$(\neg T \wedge M)$ ***or*** both not T and M

(The whole of the second analysis is parenthesized to make the comparison easier.)

The order of the words expressing negation and conjunction in the two English sentences corresponds exactly to their order in the analysis written using English notation. In particular, the word both can be seen to function in the English sentences, as it does in the analysis, to mark the beginning of the scope of a conjunction and thus to indicate whether the word not applies to the whole conjunction or only a part. Of course, things do not always work out this neatly in English, but the use of both after not is an important way of indicating exactly what is being denied. Emphasis is another way of indicating the scope of negation, and an emphasized both—as in It was not *both* hot and humid—can be particularly effective.

The real significance of negated conjunction lies in the way it modifies while combining, allowing us to say that at least one of the two components of the not-both form is false. The sentence It was not both hot and humid is false only when the components It was hot and It was humid are both true, so it leaves open every possibility in which at least one of them is false. And this is something we could not do by modifying the components separately and asserting each. On the other hand, a conjunction one or both of whose components is negative merely combines by adding content, and we could convey the same information by asserting the conjuncts separately.

While the not-both is the important new idea, conjunction of possibly negative components sometimes captures what we want to say; and there is a construction in English that seems designed to produce a logical form of this sort. The sentence It was humid but it wasn't hot could be rephrased as It was humid but not hot and thus as It was humid without being hot. So this last sentence, too, can be understood as a conjunction (i.e., as $M \wedge \neg T$ or both M and not T). Now without (in this use of the word) is a preposition, not a conjunction, so what follows it will not have the form of a sentence. But the object of without can be a nominalized predicate or nominalized sentence rather than an ordinary noun or noun phrase, and just about anything of the form $\varphi \wedge \neg \psi$ (which we will refer to as the *but-not form*) can be paraphrased using without. For example, Sue listened but didn't respond can be paraphrased as Sue listened without responding, and Ann walked in but Bill didn't see her could be paraphrased as Ann walked in without Bill seeing her. And, even when the object of without is an ordinary noun or noun phrase (rather than a nominalized predicate or sentence), the effect of without is often the same as that of a but-not form. Thus Tom left without his coat could be paraphrased as Tom left but didn't take his coat and thus analyzed as Tom left ∧ ¬ Tom took his coat. Of course, we have had to supply the verb take here, and we cannot expect any one pattern of paraphrase to work in all cases where without has an ordinary noun or noun phrase as its object.

Since this use of without is not a grammatical conjunction, it does not introduce a second main verb; and this makes it especially convenient when we want to negate a but-not form. For the easiest way to express the negation of a whole sentence is to apply not to a single main verb. Suppose we wish to say something with the following form:

$\neg$ (it will fall $\wedge$ $\neg$ it will be pushed)

not both it will fall and not it will be pushed

We might manage by expressing the three connectives one by one, ending with something like It won't both fall and not be pushed, where we have contrived a single conjoined predicate incorporating negation. But any such sentence is likely to be rather awkward. The natural way of making the claim analyzed above is to use It won't fall without being pushed. Accordingly, let us refer to the form $\neg (\varphi \wedge \neg \psi)$ as the *not-without form*.

Of course, it is also possible to conjoin sentences both of which are negations. Indeed, It was not hot and not humid is sometimes an accurate description of the weather. We would analyze this symbolically as $\neg T \wedge \neg M$ or both not T and not M. It will, at least for the time being, be convenient to have a label for the form $\neg \varphi \wedge \neg \psi$, too; and the natural one is *not-and-not form*. Although this is an important sort of truth-functional compound, we will see another way of expressing it in the next chapter that is closer to the grammatical form usually taken by such compounds in English. For the more idiomatic way of say that is not hot and also not humid is with the sentence It is neither hot nor humid. We noted earlier that this sentence can be seen as a negation of It is either hot or humid, and its analysis along those lines will await our account of the word or. But, until we have that, the not-and-not form can serve as an analysis of neither-nor sentences since it has the right truth conditions.

This way of analyzing neither-nor sentences is not the only case where conjunction and negation can be used to analyze sentences that we will be able to analyze in a different and more direct way later. For example, many if-then sentences can be analyzed using the not-without form (though doing so may be jarring due to differences in implicatures). But this is just a special case of something that was noted earlier: any truth-functional compound can be expressed using conjunction and negation alone.

To see this, suppose the effect of some connective on the truth conditions of a sentence can be captured in a truth table—that is, suppose the connective is truth-functional. The force of a sentence formed by such a connective is to deny that the actual state (or history) of the world is described by any of the rows of the table in which the sentence is false. Now the description of the state of the world offered by a given row can be captured by a run-on conjunction that affirms or denies each component in turn. For example, knowing that φ is assigned **T** and ψ is assigned **F** comes to the same thing as knowing that the sentence $\varphi \wedge \neg \psi$ is true. As a result, the compound sentence as a whole is equivalent to a conjunction of the denials of the sentences corresponding to each row in which the sentence is false. (At least this is so, if there are any such rows; otherwise, the sentence is a formal tautology and is equivalent to any other formal tautology, for example, $\neg (\varphi \wedge \neg \varphi)$.) This argument applies no matter how many components the connective applies to and no matter what form the truth table takes. For this reason, conjunction and negation are said to form a *truth-functionally complete* set of connectives.

To take a particular case, a compound with the table below can be thought of as saying that φ and ψ are not both truth and also that they are not both false, so it will be equivalent to $\neg (\varphi \wedge \psi) \wedge \neg (\neg \varphi \wedge \neg \psi)$.

φ	ψ	
T	T	F
T	F	T
F	T	T
F	F	F

An English sentence whose grammatical form is close to this form—such as Sam didn't eat both pie and cake, but he also didn't eat neither—will be very cumbersome, and there are likely to be more idiomatic ways of saying the same thing whose most natural analyses would be different. But it is still important to note that it is possible to say this sort of thing by putting the sentences Sam ate pie and Sam ate cake together using conjunction and negation alone since it shows that the other expressions for this truth function do not introduce any fundamentally new logical ideas.

3.1.5. Some sample analyses

We will conclude this discussion with several examples illustrating the issues we have discussed. First, consider a case that is entirely straightforward.

It isn't warm out
¬ it's warm out

¬ W
not W

W: it's warm out

A second example shows that uncovering even a simple form can require some thought and a paraphrase.

No one saw anyone enter the building
¬ someone saw someone enter the building

¬ S
not S

S: someone saw someone enter the building

Care is needed in distinguishing not-both forms from not-and-not forms. Everyone understands the distinction quite well intuitively, but it is easy to get tripped up when you are first learning to make this understanding explicit. Compare the following.

Britain and France won't both vote	Britain and France both won't vote
¬ Britain and France will both vote	Britain won't vote ∧ France won't vote
¬ (Britain will vote ∧ France will vote)	¬ Britain will vote ∧ ¬ France will vote
¬ (B ∧ F)	¬ B ∧ ¬ F
not both B and F	both not B and not F

B: Britain will vote; F: France will vote

The negation of a conjunction is not the same as a conjunction of negations. The second form is also the way we would analyze Neither Britain nor France will vote.

The scope of negation is one respect in which English sentences are often ambiguous, and it is not hard to find examples that people will interpret differently. For example, you may find it possible to understand the second sentence above as a denial of Britain and France will both vote—i.e., as equivalent to the first. The first seems unambiguous, but other sentences in which not appears before both are less clear. For example, it might be possible to understand Tom didn't like both the service and the price to say that he liked neither (if you have trouble understanding it to say anything *but* that, try reading it with an emphasis on both).

Finally, here is a somewhat longer example.

Al didn't get to both the meeting and the party without missing both the game and the movie
¬ Al got to both the meeting and the party without missing both the game and the movie
¬ (Al got to both the meeting and the party ∧ ¬ Al missed both the game and the movie)
¬ ((Al got to the meeting ∧ Al got to the party) ∧ ¬ (Al missed the game ∧ Al missed the movie))
¬ ((Al got to the meeting ∧ Al got to the party) ∧ ¬ (¬ Al got to the game ∧ ¬ Al got to the movie))

¬ ((T ∧ P) ∧ ¬ (¬ G ∧ ¬ V))
not both T and P and not both not G and not V

G: Al got to the game; P: Al got to the party; T: Al got to the meeting; V: Al got to the movie

The final step of analyzing X missed Y as contradictory to X got to Y is not crucial at this point in the course. While it is important to exhibit as much logical structure as possible, we end up with four logically independent sentences whether we carry out the final step or not. However, we will later go on to press analyses below the level of sentences, and this sort of step will then be of value since it leads us to four components that differ only in the object of the preposition to and therefore can be analyzed in a way that re-uses vocabulary.

Glen Helman 03 Aug 2010

3.1.s. Summary

1 Negation is an operation associated with the English word not. It generates a compound sentence from a single component, so it is a connective that serves to modify a sentence rather than to combine sentences. The not symbol $\neg$ is our notation for negation. As English notation for $\neg\,\varphi$, we use not φ.

2 A sentence and its negation cannot be both true (they are mutually exclusive) and cannot be both false (they are jointly exhaustive); in short, they must have different truth values (they are contradictory). Each leaves open the possibilities the other rules out and rules out the possibilities the other leaves open. This means that negation, like conjunction, has a truth table; in other words it is a truth-functional connective. Not all connectives are truth-functional. Truth-functional logic is the branch of logic which studies those that are, but there are branches of logic—such as tense logic and modal logic—in which non-truth-functional connectives are studied.

3 Negation appears in English not only in connection with the word not but also with negative prefixes (though such a prefix does not always mark negation because it does not always produce a sentence that is contradictory to the original). Negation also appears with uses of no in phrases of the form no X, uses that can often be treated as the negation of at least one or some. The same sort of treatment is usually what is required when not appears along with the word any (though such sentences usually must be rephrased when not is removed). By negating a negation, we can produce a double negation, but this undoes the negation rather than generating a logical form with new properties.

4 The really new ideas come with the negation of conjunctions, but conjunctions whose components may involve negation also provide important forms of expression. A number of forms are shown below, with labels that suggest the sort of English sentences they serve to analyze:

not-both form	$\neg\,(\varphi \land \psi)$	not both φ and ψ
not-but form	$\neg\,\varphi \land \psi$	both not φ and ψ
but-not form	$\varphi \land \neg\,\psi$	both φ and not ψ
not-and-not form	$\neg\,\varphi \land \neg\,\psi$	both not φ and not ψ
not-without form	$\neg\,(\varphi \land \neg\,\psi)$	not both φ and not ψ

That the last is the denial of the third reflects the fact that without can be used to express a but-not form. Also neither-nor can be used to express a not-and-not form. More generally, negation and conjunction form a truth-functionally complete set of connectives in the sense that any truth-functional compound can be expressed using them alone.

Glen Helman 03 Aug 2010

3.1.x. Exercise questions

1. Analyze each of the following sentences in as much detail as possible.

- **a.** The soup was hot but not too hot, and thick but not too thick.
- **b.** The equipment isn't here and it's unlikely to arrive soon.
- **c.** No one answered the phone even though it rang 10 times.
- **d.** The alarm must have gone off, but Ted didn't hear anything.
- **e.** They won't both meet the deadline and stay within the budget.
- **f.** They won't meet the deadline, but they will stay within the budget.
- **g.** They won't meet the deadline, and they won't stay within the budget.
- **h.** Tod shut off the alarm without waking up.
- **i.** They won't meet the deadline without going over the budget.
- **j.** Larry joined in, but not without being coaxed.
- **k.** Ann liked the movie, but neither Bill nor Carol did.

2. Restate each of the forms below, putting English notation into symbols and vice versa. Indicate the scope of connectives in the result by underlining.

- **a.** $\neg \neg (A \wedge B)$
- **b.** $\neg (\neg A \wedge B)$
- **c.** both not A and both not B and C
- **d.** both not both A and B and not C

3. Synthesize idiomatic English sentences that express the propositions that are associated with the logical forms below by the intensional interpretations that follow them.

- **a.** $C \wedge \neg F$
  C: it was cold; F: there was frost
- **b.** $\neg S \wedge (H \wedge I)$
  H: Sue heard a crash; I: Sue went to investigate; S: someone saw the accident
- **c.** $(D \wedge N) \wedge \neg P$
  D: it was a design; N: it was new; P: it pleased someone
- **d.** $\neg (I \wedge N)$
  I: we'll win in Iowa; N: we'll win in New York
- **e.** $\neg I \wedge N$
  I: we'll win in Iowa; N: we'll win in New York
- **f.** $\neg (I \wedge \neg L)$
  I: we'll win in Iowa; L: we'll lose in New York

4. Complete the following truth tables. That is, calculate truth values for all components of the forms below using the extensional interpretation provided on the left in each case.

a.

A	B	C	$A \wedge \neg (B \wedge C)$
T	F	F	

b.

A	B	C	$A \wedge (\neg B \wedge C)$
T	F	F	

c.

A	B	C	D	$(\neg A \wedge \neg B) \wedge (\neg (A \wedge C) \wedge D)$
F	T	T	T	

Glen Helman 03 Aug 2010

3.1.xa. Exercise answers

1. **a.** The soup was hot but not too hot ∧ the soup was thick but not too thick

(the soup was hot ∧ the soup was not too hot) ∧ (the soup was thick ∧ the soup was not too thick)

(the soup was hot ∧ ¬ the soup was too hot) ∧ (the soup was thick ∧ ¬ the soup was too thick)

(H ∧ ¬ T) ∧ (K ∧ ¬ O)

both both H and not T and both K and not O

H: the soup was hot; K: the soup was thick; O: the soup was too thick; T: the soup was too hot

b. The equipment isn't here ∧ the equipment is unlikely to arrive soon

¬ the equipment is here ∧ ¬ the equipment is likely to arrive soon

¬ H ∧ ¬ S

both not H and not S

H: the equipment is here; S: the equipment is likely to arrive soon

c. No one answered the phone ∧ the phone rang 10 times

¬ someone answered the phone ∧ the phone rang 10 times

¬ A ∧ R

both not A and R

A: someone answered the phone; R: the phone rang 10 times

d. The alarm must have gone off ∧ Ted didn't hear anything

The alarm must have gone off ∧ ¬ Ted heard something

A ∧ ¬ H

both A and not H

A: the alarm must have gone off; H: Ted heard something

e. ¬ they will both meet the deadline and stay within the budget

¬ (they will meet the deadline ∧ they will stay within the budget)

¬ (D ∧ B)

not both D and B

B: they will stay within the budget; D: they will meet the deadline

f. They won't meet the deadline ∧ they will stay within the budget

¬ they will meet the deadline ∧ they will stay within the budget

¬ D ∧ B

both not D and B

B: they will stay within the budget; D: they will meet the deadline

g. They won't meet the deadline ∧ they won't stay within the budget

¬ they will meet the deadline ∧ ¬ they will stay within the budget

¬ D ∧ ¬ B

both not D and not B

B: they will stay within the budget; D: they will meet the deadline

h. Tod shut off the alarm ∧ ¬ Tod woke up

A ∧ ¬ W

both A and not W

A: Tod shut off the alarm; W: Tod woke up

i. ¬ they will meet the deadline without going over the budget

¬ (they will meet the deadline ∧ ¬ they will go over the budget)

¬ (D ∧ ¬ G)

not both D and not G

D: they will meet the deadline; G: they will go over the budget

j. Larry joined in ∧ Larry did not join in without being coaxed

Larry joined in ∧ ¬ Larry joined in without being coaxed

Larry joined in ∧ ¬ (Larry joined in ∧ ¬ Larry was coaxed)

J ∧ ¬ (J ∧ ¬ C)

both J and not both J and not C

C: Larry was coaxed; J: Larry joined in

This is equivalent to J ∧ ¬ ¬ C and also to J ∧ C, but the analysis shown is closer to the form of the English.

k. Ann liked the movie ∧ neither Bill nor Carol liked the movie

Ann liked the movie ∧ (¬ Bill liked the movie ∧ ¬ Carol liked the movie)

$$A \land (\neg B \land \neg C)$$

both A and both not B and not C

A: Ann liked the movie; B: Bill liked the movie; C: Carol liked the movie

The alternative (and equivalent) analysis as A ∧ ¬ E (where E: either Bill or Carol liked the movie) is closer to the English but it is less satisfactory because it displays less structure. The next chapter will give us the means carry this sort of analysis further by analyzing E as a compound of B and C.

2. **a.** not not both A and B

b. not both not A and B

c. ¬ A ∧ (¬ B ∧ C)

d. ¬ (A ∧ B) ∧ ¬ C

3. **a.** It was cold ∧ ¬ there was frost

It was cold ∧ there was no frost

It was cold, but there was no frost

b. ¬ someone saw the accident ∧ (Sue heard a crash ∧ Sue went to investigate)

No one saw the accident ∧ Sue heard a crash and went to investigate

No one saw the accident, but Sue heard a crash and went to investigate

c. (it was a design ∧ it was new) ∧ ¬ it pleased someone

It was a new design ∧ it pleased no one

It was a new design, and it pleased no one

d. ¬ (we'll win in Iowa ∧ we'll win in New York)

¬ (we'll win in both Iowa and New York)

We won't win in both Iowa and New York

e. ¬ we'll win in Iowa ∧ we'll win in New York

We won't win in Iowa ∧ we'll win in New York

We won't win in Iowa, but we'll win in New York

f. ¬ (we'll win in Iowa ∧ ¬ we'll lose in New York)

¬ (we'll win in Iowa without losing in New York)

We won't win in Iowa without losing in New York

4. Numbers below the tables indicate the order in which values were computed.

a.

A	B	C	A	∧	¬	(B	∧	C)
T	F	F		Ⓣ	T		F	
				3	2		1	

b.

A	B	C	A	∧	(¬	B	∧	C)
T	F	F		Ⓕ	T		F	
				3	1		2	

[Note that, while in **a**, it is the value under the ¬ that is used in calculating the value of the main conjunction, in **b** it is the value under the second ∧; this is due to the change in relative scope of these two connectives.]

c.

A	B	C	D	(¬	A	∧	¬	B)	∧	(¬	(A	∧	C)	∧	D)
F	T	T	T	T		F	F		Ⓕ	T		F		T	
				1		2	1		4	2		1		3	

Glen Helman 03 Aug 2010

3.2. *Reductio* arguments: refuting suppositions

3.2.0. Overview

Since negating a sentence changes what it says into the contradictory opposite, the role of negation in deductive reasoning is quite different from the role of conjunction; and rules for negation will focus on the rejection of sentences rather than the extraction and assembly of information.

3.2.1. The duality of premises and alternatives
: The deductive properties of negation rest on ties between the relation between premises and alternatives on the one hand and the relation between a sentence and its negation on the other.

3.2.2. Drawing negative conclusions
: The basic form of argument for a negative conclusion establishes a relation of exclusion, and it does so by a reduction to absurdity.

3.2.3. Some examples
: An account of the role of negation as a conclusion does not capture all its deductive properties, but many of the most typical sorts of negative argumentation do follow.

Glen Helman 03 Aug 2010

3.2.1. The duality of premises and alternatives

The law alternatives *via* assumptions tells us that, when sentences φ and $\overline{\varphi}$ are contradictory, having one as a premise comes to the same thing as having the other as a conclusion—that is,

$$\Gamma \vDash \varphi, \Delta \text{ if and only if } \Gamma, \overline{\varphi} \vDash \Delta$$

If we apply this to the contradictories φ and $\neg\,\varphi$, we get a pair of principles

$$\Gamma \vDash \neg\,\varphi, \Delta \text{ if and only if } \Gamma, \varphi \vDash \Delta$$
$$\Gamma, \neg\,\varphi \vDash \Delta \text{ if and only if } \Gamma \vDash \varphi, \Delta$$

where we get the second by reversing the contradictory pair. The two together tell us that having a negation as either a premise or alternative comes to the same thing as having the unnegated sentence in the opposite role (where the opposition in question is the duality mentioned in 1.4.7).

We do not study conditional exhaustiveness directly, and we use of the basic law for conditional exhaustiveness mainly to exchange alternatives for premises so that a claim of conditional exhaustiveness may be converted into a claim of entailment. But suppose we apply it to entailment instead; that is, suppose we begin with only a single alternative (so the set Δ is empty). In this case, when φ and $\overline{\varphi}$ are contradictory, we can say that

$$\Gamma \vDash \varphi \text{ if and only if } \Gamma, \overline{\varphi} \vDash$$

where the right-hand side says that $\overline{\varphi}$ is inconsistent with (or is excluded by) Γ. When we express that inconsistency as the validity of a *reductio* argument, we get the following principle:

$$\text{if } \varphi \text{ and } \overline{\varphi} \text{ are contradictory, then } \Gamma \vDash \varphi \text{ if and only if } \Gamma, \overline{\varphi} \vDash \bot$$

And this will be the basis for our account of negation.

We get our basic principles for negation by applying this principle to the case of negation by choosing the contradictory pair as a sentence and its negation, both in that order and its reverse. Turning the second if and only if principle around so that clause concerning negation comes first, the two principles are these:

LAW FOR NEGATION AS A CONCLUSION. $\Gamma \vDash \neg\,\varphi$ if and only if $\Gamma, \varphi \vDash \bot$.
LAW FOR NEGATION AS A PREMISE. $\Gamma, \neg\,\varphi \vDash \bot$ if and only if $\Gamma \vDash \varphi$.

Although these principles are dual in something like the way that the earlier pair for conditional exhaustiveness were, each has a rather different significance. The first captures the core properties of negation while the second is closely tied to the equivalence of $\neg\,\neg\,\varphi$ with φ (which, as was noted in 3.1.3

, is about as controversial as anything gets in logic). Also, while the first will provide us with straightforward ways of planning for negative goals and carrying out these plans, the second gives an account of the role of negative premises only in the context of *reductio* arguments and, for this reason, has a less straightforward implementation as a derivation rule. We will go on to explore the implementation of the first now and postpone a discussion of the second until 3.3.

Glen Helman 04 Aug 2010

3.2.2. Drawing negative conclusions

The law for negation as a conclusion

$$\Gamma \vDash \neg \varphi \text{ if and only if } \Gamma, \varphi \vDash \bot$$

describes the conditions under which an entailment of the form $\Gamma \vDash \neg \varphi$ holds. An example may help in thinking about this law. The argument

Ann and Bill were not both home without the car being in the driveway
Ann was home but the car was not in the driveway

Bill was not home

is valid and seeing that it is valid comes to the same thing as seeing that Bill could not have been home if the premises are true. But to see this is to see that the claim Bill was home is excluded by the premises of the argument. So the negative conclusion of this argument is valid because the conclusion denies something that is excluded by the premises.

This connection between validity and exclusion is just what the law for negation as a conclusion states. For a *reductio* entailment $\Gamma, \varphi \vDash \bot$ is the way we capture exclusion in terms of entailment: Γ excludes φ if adding φ to Γ would enable us to reach an absurd conclusion. And the law tells us that we can validly conclude a negation $\neg \varphi$ when we can reduce to absurdity the result of adding φ to the premises Γ. So we can say that the example above is valid because due to the inconsistency of the three sentences

Ann and Bill were not both home without the car being in the driveway
Ann was home but the car was not in the driveway
Bill was home

We can reduce these claims to absurdity by noting that the second and third together imply Ann and Bill were both home without the car being in the driveway and that this is what the first denies.

Although this reduction to absurdity shows the inconsistency of the full set of sentences from which we draw the absurd conclusion, we focus attention on the last one of them to draw a negative conclusion from the first two. And in general, the entailment $\Gamma, \varphi \vDash \bot$ shows the inconsistency of the full set containing the members of Γ together with φ, but we focus attention on φ when we say it is excluded by, or is inconsistent with, Γ. We can focus attention on a single sentence when speaking of reduction to absurdity itself by saying that the argument Γ, φ / ⊥ reduces φ to absurdity given Γ. And this allows us to restate the law for negation as a conclusion in another way: we can validly conclude a negation ¬ φ from premises Γ when we can reduce φ to absurdity given the premises Γ. In the example above, we reduced Bill was home to absurdity given the two premises of the original argument.

As a rule for sequent proofs, the principle for negation as a conclusion would take the following form:

$$\text{neg. as concl.}\ \frac{\Gamma, \varphi \vDash \bot}{\Gamma \vDash \neg\, \varphi}$$

And this shows the effect we want a corresponding derivation rule to have on a gap in a derivation. There is no branching, but a premise is added in moving up from the bottom and the required conclusion is strengthened to ⊥.

To implement this idea in derivations, we must add φ as a further resource in the child gap. Unlike resources added through Ext, this added resource will generally go beyond information contained in the premises. It is a genuine addition to the claims made by the premises, amounting to a further assumption for the purposes of the argument. Such further assumptions are often called *suppositions* and the verb suppose is used to introduce them when putting this sort of deductive reasoning into words. Suppositions can have a variety of roles in deductive reasoning. In the rules Lemma and LFR of 2.4, a lemma is introduced as a supposition in one gap of a derivation. In those rules this supposition represents a resource that we have on loan, a loan that is paid if we are able to prove the lemma in another gap. When we suppose φ in order to prove ¬ φ, we make the supposition in order to refute it by reducing it to absurdity. That is, we make the supposition in order to consider a possibility, and we go on to rule out the possibility on the basis of other assumptions. We will encounter still other uses of suppositions in later chapters.

The rule that implements this idea in derivations will be called *Reductio Ad Absurdum* or RAA. It is shown in Figure 3.2.2-1.

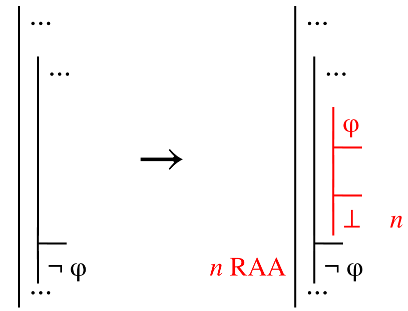


Fig. 3.2.2-1. Developing a derivation by planning for a negation at stage *n*.

This rule leads us to develop a gap by adding a supposition and, at the same time, changing our goal to ⊥. The part of the derivation these changes affect is marked by a scope line, and the added resource is marked off at the top by a horizontal line.

If we state this rule for tree-form proofs, it takes the following form (which you should compare to the analogous diagram for the rule Lem of 2.4.1):

$$\begin{array}{c} \not\varphi \\ \vdots \end{array}$$

$$\text{RAA}\ \frac{\bot}{\neg\, \varphi}$$

This shows a pattern of argument in which we conclude ¬ φ from the premise ⊥. But that description would apply also to the rule EFQ, so it does not capture all that is going on here. The conclusion ¬ φ is, in general, weaker than ⊥. And the rule for negation as a conclusion tells us that the particular way it is weaker licenses us to drop φ from our assumptions. Since the conclusion rules out no case where φ is false, it need no longer depend on an assumption φ that rules out such cases.

As with other rules, the form of RAA in tree-form proofs explains the numerical annotations for it in derivations. A stage number is placed to the right of ⊥, since it is the true premise from which ¬ φ is concluded. The supposition φ is not a premise but plays a different role so no stage number is added to its right (though one might appear later if it is exploited inside the gap as it develops further). The fact that the supposition is discharged when we draw a conclusion from ⊥ is shown in the derivation simply by the fact that its scope line ends with ⊥.

Once we have begun a *reductio* argument, we have ⊥ as our goal and we must look for ways of reaching it. The only way we have in our rules so far is

QED, but that requires that we have ⊥ among our resources. While it is, of course, possible that our new supposition is ⊥ or that ⊥ was already among our resources, we would not expect this to happen in general. Usually, we will need to make use of both the supposition and the pre-existing resources and make use of some negative claims among them. Our full discussion of the use of negative resources will come only in 3.3, but the core principle for using such resources is one we can consider now.

One of the traditional laws of logic is the *law of non-contradiction*. This is sometimes referred to also as the “law of contradiction” when the focus is simply on the fact that it is a law for concluding something from a contradictory pair rather than the fact that what we conclude is that they cannot be both true. We know it as the principle that $\neg\varphi$ and φ are mutually exclusive—or, in the form most relevant at the moment, that $\neg\varphi, \varphi \vDash \bot$.

This idea lies behind a pattern of argument that we will call *Non-contradiction* or Nc:

$$\text{Nc}\ \frac{\neg\varphi \qquad \varphi}{\bot}$$

This pattern of argument will appear in derivations as a way of completing a *reductio* argument:

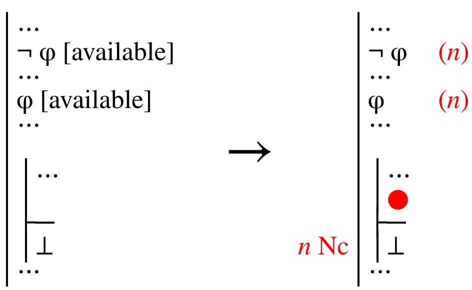


Fig. 3.2.2-2. Closing the gap of a *reductio* argument one of whose resources negates another.

Notice that, as with other rules that close gaps, the resources required to apply this need only be available and they are marked with parenthesized stage numbers. The latter point is moot, as it was with QED and EFQ, since the gap closes. And, in a way, the possibility of using available but inactive resources is moot also. Once we have the further rules of 3.3, we will need this rule only when φ is an unanalyzed component (though it will be usable and useful in other cases, too). And we will never have rules for exploiting unanalyzed components or their negations, so such resources will be active whenever they are available.

Glen Helman 11 Sep 2010

3.2.3. Some examples

Here is a derivation that uses the rules RAA and Nc:

```
         │ A ∧ ¬ C               1
         ├──
1 Ext    │ A
1 Ext    │ ¬ C                   (5)
         │ │ B ∧ (C ∧ ¬ D)       3
         │ ├──
3 Ext    │ │ B
3 Ext    │ │ C ∧ ¬ D             4
4 Ext    │ │ C                   (5)
4 Ext    │ │ ¬ D
         │ │ ●
         │ ├──
5 Nc     │ │ ⊥                   2
         ├──
2 RAA    │ ¬ (B ∧ (C ∧ ¬ D))
```

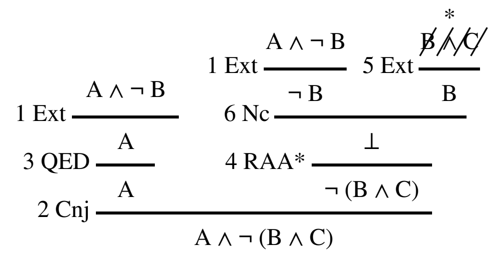
One feature of this derivation will now be typical: it is not possible to have all uses of Ext at the beginning of the derivation since this rule may be used to exploit suppositions.

Of course, we might have used RAA at the first stage before applying Ext. But the following derivation shows that even this degree of grouping of uses of Ext will not always be possible.

```
         │ A ∧ ¬ B               1
         ├──
1 Ext    │ A                     (3)
1 Ext    │ ¬ B                   (6)
         │ │ ●
         │ ├──
3 QED    │ │ A                   2
         │ │ │ B ∧ C             5
         │ │ ├──
5 Ext    │ │ │ B                 (6)
5 Ext    │ │ │ C
         │ │ │ ●
         │ │ ├──
6 Nc     │ │ │ ⊥                 4
         │ ├──
4 RAA    │ │ ¬ (B ∧ C)           2
         ├──
2 Cnj    │ A ∧ ¬ (B ∧ C)
```

We might have waited until after the supposition B ∧ C was made before applying Ext to the initial premise; but, by then, there would be two gaps and the first premise would have to be exploited in each in order for them to close. In general, it is wise (though not necessary) to apply Ext to a conjunction as soon as it appears as a resource, but conjunctions may continue to appear as resources from time to time as a derivation develops.

The tree-form proof corresponding to the second of these derivations takes the following form:



Notice the asterisk above the discharged supposition B ∧ C. This ties the supposition to the rule that discharges it in the way the scope line of the supposition does in a derivation.

Now let's look at the sort of derivation we might give for the argument that began 3.2.2. We can analyze the first premise of that argument as follows:

Ann and Bill were not both home without the car being in the driveway

¬ Ann and Bill were both home without the car being in the driveway

¬ (Ann and Bill were both home ∧ ¬ the car was in the driveway)

¬ ((Ann was home ∧ Bill was home) ∧ ¬ the car was in the driveway)

¬ ((A ∧ B) ∧ ¬ C)

not both both A and B and not C

A: Ann was home; B: Bill was home; C: the car was in the driveway

So the full argument takes the form:

¬ ((A ∧ B) ∧ ¬ C)
A ∧ ¬ C
―――――――
¬ B

The negative first premise is crucial for the argument, but we have no way of using it at the moment without having the compound it negates as a resource. To get that compound—i.e., (A ∧ B) ∧ ¬ C—as a resource, we need to use Adjunction to build its first conjunct and then the full compound.

	$\neg((A \wedge B) \wedge \neg C)$	(6)
	$A \wedge \neg C$	2
2 Ext	A	(4)
2 Ext	$\neg C$	(5)
	B	(4)
4 Adj	$A \wedge B$	X,(5)
5 Adj	$(A \wedge B) \wedge \neg C$	X,(6)
	●	
6 Nc	$\bot$	3
3 RAA	$\neg B$	

The need to use Adjunction in cases like this will end when we get the further rules of the next section, but it will sometimes still be a natural approach to establishing an entailment.

Now let's see what the derivation looks like if we replace the symbolic sentences by the actual English sentences they analyze:

	Ann and Bill were not both home without the car being in the driveway	(6)
	Ann was home but the car was not in the driveway	2
2 Ext	Ann was home	(4)
2 Ext	the car was not in the driveway	(5)
	Bill was home	(4)
4 Adj	Ann and Bill were both home	X,(5)
5 Adj	Ann and Bill were both home without the car being in the driveway	X,(6)
	●	
6 Nc	$\bot$	3
3 RAA	Bill was not home	

In a stretch of explicit deductive argumentation in English, various sorts of connecting language would be used to get the effect of the lines and annotations that structure this derivation. Although this is not the sort of entailment where such an explicit argument would ordinarily be given, if one were offered, it might run something like this:

> We assume that Ann and Bill were not both home without the car being in the driveway and also that Ann was home but the car was not in the driveway. So we know that Ann was home. And we also know that the car was not in the driveway.
>
> Now suppose (for the sake of *reductio*) that Bill was home. It would follow that Ann and Bill were both home. And then we would know that Ann and Bill were both home without the car being in the driveway. But that contradicts one of our initial assumptions.
>
> So we can conclude that Bill was not home.

The modal verb would has been used here in the *reductio* argument of the second paragraph to emphasize that the situation being described need not be a real one. It is possible to go further in that direction by phrasing the supposition itself as Suppose that Bill were home; but it is also possible to let the verb suppose suffice to show that what follows is not a consequence of the initial premises.

Glen Helman 03 Aug 2010

3.2.s. Summary

1 The basic law for exhaustiveness says that having one of a pair of contradictory sentences as a premises comes to the same thing as having the other as an alternative. This does not apply to entailment directly, but we can consider a special case which says that one of a pair of contradictory sentences is entailed by a set if and only if the other is inconsistent with that set. Since a sentence and its negation are contradictories, this gives us a pair of principles, laws for negation as a premise and as a conclusion.

2 Inconsistency is established by a *reductio* argument. In a derivation, this will be associated with a gap that has ⊥ as its goal. In order to show a sentence inconsistent with our premises, we add it as a further assumption in the *reductio* argument. This further assumption may be referred to as a supposition of this argument to distinguish it from the premises with which we hope to show it inconsistent. The rule implementing this idea is Reductio ad Absurdum (RAA). To actually reach the goal of ⊥, we add a rule allowing us to close a gap when a sentence and its negation are among the resources. This rule is Non-contradiction (Nc) and is named after the traditional law of non-contradiction.

3 The use of suppositions means that we will no longer always be able to group all uses of Ext at the beginning of a derivation. A more temporary complication is the need to use Adj to form a sentence contradictory to a negated conjunction, something that will be handled by a rule introduced in the next section.

Glen Helman 03 Aug 2010

3.2.x. Exercise questions

1. Use derivations to establish each of the claims of entailment shown below. Notice that **c** is a claim of tautologousness; it requires a derivation without initial assumptions. All the resources used in a such a derivation will come from suppositions.

a. $\neg A \vDash \neg (A \wedge B)$

b. $\neg B \vDash \neg (A \wedge B) \wedge \neg (B \wedge C)$

c. $\vDash \neg (A \wedge \neg A)$

d. $J \wedge C \vDash J \wedge \neg (J \wedge \neg C)$ (see exercise **1j** of 3.1.x)

2. Use derivations to establish each of the claims of entailment shown below. You will need to introduce lemmas to exploit the negated compounds that appear as premises. For most, Adj is enough; but, for the last, you will need to use the rule LFR introduced in 2.4.

a. $\neg (A \wedge B), A \vDash \neg B$

b. $\neg (A \wedge \neg B), \neg B \vDash \neg A$

c. $A, \neg (A \wedge B), \neg (A \wedge C) \vDash \neg B \wedge \neg C$

d. $\neg (A \wedge B), \neg (C \wedge \neg B) \vDash \neg (A \wedge C)$

We have too limited a group of rules at this point for the exercise machine to be useful.

Glen Helman 03 Aug 2010

3.2.xa. Exercise answers

1. **a.**

	¬ A	(3)
	│ A ∧ B	2
2 Ext	│ A	(3)
2 Ext	│ B	
	│ ●	
3 Nc	│ ⊥	1
1 RAA	¬ (A ∧ B)	

b.

	¬ B	(4),(7)
	│ │ A ∧ B	3
3 Ext	│ │ A	
3 Ext	│ │ B	(4)
	│ │ ●	
4 Nc	│ │ ⊥	2
2 RAA	│ ¬ (A ∧ B)	1
	│ │ B ∧ C	6
6 Ext	│ │ B	(7)
6 Ext	│ │ C	
	│ │ ●	
7 Nc	│ │ ⊥	5
5 RAA	│ ¬ (B ∧ C)	1
1 Cnj	¬ (A ∧ B) ∧ ¬ (B ∧ C)	

c.

	│ A ∧ ¬ A	2
2 Ext	│ A	(3)
2 Ext	│ ¬ A	(3)
	│ ●	
3 Nc	│ ⊥	1
1 RAA	¬ (A ∧ ¬ A)	

d.

	J ∧ C	1
1 Ext	J	(3)
1 Ext	C	(6)
	│ ●	
3 QED	│ J	2
	│ │ J ∧ ¬ C	5
5 Ext	│ │ J	
5 Ext	│ │ ¬ C	(6)
	│ │ ●	
6 Nc	│ │ ⊥	4
4 RAA	│ ¬ (J ∧ ¬ C)	2
2 Cnj	J ∧ ¬ (J ∧ ¬ C)	

2. **a.**

	¬ (A ∧ B)	(3)
	A	(2)
	│ B	(2)
2 Adj	│ A ∧ B	X,(3)
	│ ●	
3 Nc	│ ⊥	1
1 RAA	¬ B	

b.

	¬ (A ∧ ¬ B)	(3)
	¬ B	(2)
	│ A	(2)
2 Adj	│ A ∧ ¬ B	X,(3)
	│ ●	
3 Nc	│ ⊥	1
1 RAA	¬ A	

c.

	A	(3),(6)
	¬ (A ∧ B)	(4)
	¬ (A ∧ C)	(7)
	│ │ B	(3)
3 Adj	│ │ A ∧ B	X,(4)
	│ │ ●	
4 Nc	│ │ ⊥	2
2 RAA	│ ¬ B	1
	│ │ C	(6)
6 Adj	│ │ A ∧ C	X,(7)
	│ │ ●	
7 Nc	│ │ ⊥	5
5 RAA	│ ¬ C	1
1 Cnj	¬ B ∧ ¬ C	

d.

	¬ (A ∧ B)	(6)
	¬ (C ∧ ¬ B)	(8)
	│ A ∧ C	2
2 Ext	│ A	(5)
2 Ext	│ C	(7)
	│ │ │ B	(5)
5 Adj	│ │ │ A ∧ B	X,(6)
	│ │ │ ●	
6 Nc	│ │ │ ⊥	4
4 RAA	│ │ ¬ B	3
	│ │ ¬ B	(7)
7 Adj	│ │ C ∧ ¬ B	X,(8)
	│ │ ●	
8 Nc	│ │ ⊥	3
3 LFR	│ ⊥	1
1 RAA	¬ (A ∧ C)	

3.3. Negations as premises

3.3.0. Overview

A second group of rules for negation interchanges the roles of an affirmative sentence and its negation.

3.3.1. Indirect proof

The basic principles for negation describe its role as a premise only in *reductio* arguments but a *reductio* is always available as an argument of last resort.

3.3.2. Using lemmas to complete *reductio*s

The role negative resources play will be to contradict other sentences; since what they contradict must often be introduced as a lemma, a use of lemmas is built into the rule for exploiting negative resources.

3.3.3. More examples

These new rules permit some new approaches to entailments that could be established using the last section's rule; but they also support some further entailments.

3.3.1. Indirect proof

The last section pursued consequences of the law for negation as a conclusion. The rules of this section will implement the other basic law for negation, the law for it as a premise:

$$\Gamma, \neg\,\varphi \vDash \bot \text{ if and only if } \Gamma \vDash \varphi$$

This says that a negation is $\neg\,\varphi$ inconsistent with a set Γ if and only if the sentence φ is entailed by that set.

There are several lessons we can learn from this law. First, the only-if-statement tells us that negative conclusions are not the only ones that can be established by way of *reductio* arguments, for it says that an entailment $\Gamma \vDash \varphi$ must hold if the *reductio* $\Gamma, \neg\,\varphi \vDash \bot$ holds. Further, the if-statement tells us in part that such an approach is safe, that the *reductio* is valid whenever the argument we wish to support by it is valid. But if-statement tells us more. Notice that φ is just the sort of resource that would enable us to complete a *reductio* that has $\neg\,\varphi$ as a premise. The if-claim above tells us that, if a *reductio* with $\neg\,\varphi$ as a premise can be completed at all, we would be able to validly conclude φ as a lemma—and that we could do so without using $\neg\,\varphi$ itself as a premise. This further lesson will provide the basis for exploiting negative resources. However, its full application depends on the broader use of *reductio* arguments supported by the other two lessons, and that is what we will consider first.

Here is an example of this broader use of *reductio*s. If we take No one is home to be the negation $\neg$ someone is home, the law for negation as a premise says we can rest the validity of the left-hand argument below on the validity of the right-hand argument.

If no one was out, the car was in the driveway	If no one was out, the car was in the driveway
The car wasn't in the driveway	The car wasn't in the driveway
	No one was out
Someone was out	$\bot$

The right-hand argument depends in part on the logical properties of if; but, as far as negation is concerned, it depends on only the fact that a sentence and its negation are mutually exclusive.

The fact that they are mutually exclusive also supports the entailment $\neg\,\neg\,\varphi, \neg\,\varphi \vDash \bot$. If we apply the law for negation as a premise to this entailment, we get the principle $\neg\,\neg\,\varphi \vDash \varphi$. Moreover, the latter principle can be combined with the law for negation as a conclusion to establish the law for negation as a premise. So the further logical properties of negation that are captured by the law for negation as a premise can be summarized in the principle that a double negation entails the corresponding positive claim.

This principle is one that was rejected by Brouwer in his intuitionistic mathematics. And one of his chief reasons for rejecting it was that it would allow us to draw a conclusion of the form Something has the property P when the corresponding claim Nothing has the property P was inconsistent with our premises, and that is just the sort of thing that was done in the example above. His concern with this is that it would enable us to conclude Something has the property P in cases where we were unable, even in principle, to provide an actual example of a thing with that property P. Brouwer did not object to such an argument in ordinary reasoning about the physical world (like the example above); but he held that, in reasoning concerning infinite mathematical structures, we were not reasoning about an independently existing realm of objects but instead about procedures for constructing abstract objects and that we had no business claiming the existence of such objects without having procedures enabling us to construct them. Brouwer's concerns may not lead you to question the law for negation as a premise; but they highlight the indirectness of supporting a positive conclusion by an argument concerning its denial. This aspect of these arguments is reflected in a common term for them, *indirect proofs*.

Although we will employ indirect proofs, we will need them for only a limited range of conclusions. We have other ways of planning for a goal that is a conjunction or a negation, and we can simply close a gap whose goal is $\top$. We will not adopt any rule to plan for the goal $\bot$ of a *reductio argument*. At the moment, that leaves only unanalyzed components; and, until the last chapter (where we consider the logical properties of something), those are the only goals for which we will use indirect proofs. We have often closed gaps whose goals are atomic, so we know that indirect proof is not always necessary even for such goals, but it will serve us as a last resort.

In chapter 6, we will begin to analyze sentences into components that are not sentences, and we will still use indirect proof for goals that are analyzed in that way. In anticipation of this, we will use the term *atomic* for the kind of goals to which we will apply indirect proof; and we will refer to other sentences as *non-atomic*. Until chapter 6, any sentence we analyze will be a compound formed by applying a connective to one or more sentences, so, for the time being, the atomic sentences will be the unanalyzed sentences. $\top$ and $\bot$ count as non-atomic since identifying them as Tautology and Absurdity counts as an analysis of their logical form. As a result, for the time being, the atomic sentences will be simple letters, and all other sentences will be non-atomic.

In tree-form and sequent proofs, this new form of argument will look like RAA or negation as a conclusion except that the positions of φ and ¬ φ will be reversed. The same is true of the rule implementing indirect proofs in derivations, but we will choose a name that reflects its rather different role and call it *Indirect Proof* (IP). It takes the following form:

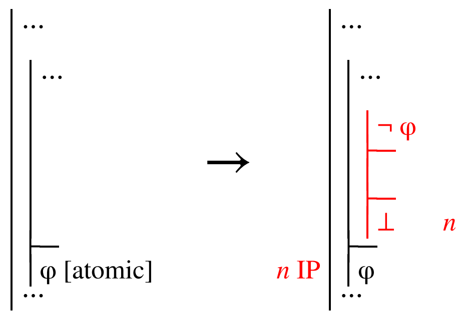


Fig. 3.3.1-1. Developing a derivation by planning for an atomic sentence at stage *n*.

Here is an example, which is related to the argument at the beginning of 3.2.2.

	¬ ((A ∧ B) ∧ ¬ C)	(4)
	A	(2)
	B	(2)
	¬ C	(3)
2 Adj	A ∧ B	X,(3)
3 Adj	(A ∧ B) ∧ ¬ C	X,(4)
	●	
4 Nc	⊥	1
1 IP	C	

This example adds to the premise Ann and Bill were not both home without the car being in the driveway further premises telling us that each of Ann and Bill was home, and we conclude that the car was in the driveway. Although the initial premises and conclusion differ from those of the argument in 3.2.2, the *reductio* argument that is set up at stage 1 here has the same resources as the *reductio* set up at stage 3 in the derivation for the argument of 3.2.2 that was given at the end of 3.2.3.

The rule IP is easily seen to be strict and safe, but we need to be more careful in assessing the decisiveness of a system using it. We will consider this question most fully in 3.4.1, but we can see the issue in outline now. Since IP introduces a sentence with one more connective than the goal it plans for, it does not reduce the quantity we have used to assess the progressiveness of other rules. But IP can be seen to be progressive nonetheless if we look progressiveness in a slightly different way. We will treat both atomic sentences and their negations as equally basic when they are resources: neither sort of resource will be exploited. And, as was noted above, we will treat ⊥ as the basic form of goal, the only one without a corresponding planning rule. Thus IP leaves us with a goal that requires no planning, and it introduces no resources that need to be exploited further. This suggests a way of looking at the distance of a proximate argument from a dead end that would allow us to say that IP reduces this quantity. This way of looking at distance from a dead end is a departure from counting connectives but only a small departure. We may count connectives except for the case of those sentences that are never exploited or planned for—i.e., atomic and negated atomic sentences as resources and ⊥ as a goal—and these sentences will be given a lower degree than all other sentences, no matter how few connectives those sentences contain.

Although IP introduces a resource that needs no exploitation, this is not to say that applying IP will eliminate the need for further exploitations; indeed, since negated compounds will be exploited only in *reductio* arguments, we will often be in a position to exploit such resources only after we have used IP. The rule it can put us in the position to use is the one we will consider next.

3.3.2. Using lemmas to complete *reductio*s

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 ⊤, ⊥, conjunction, and negation. However, to do this using only the rules we have so far, we would often need to use LFR—or, in simpler cases, Adj—to make use of negative resources. This poses no problems when we construct derivations for valid arguments, but it makes it difficult to show that an argument is not valid. LFR does not itself exploit resources, so negated compounds 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, in the case of negation, it is possible to keep track of the use of resources by way of a genuine exploitation rule, which will eliminate the need to use LFR and Adj.

The basis for this approach is one of the lessons drawn from the law of negation as a premise: if a *reductio* that has ¬ φ as a premise is valid, then φ must be a valid conclusion from the premises other than ¬ φ. That is, Γ, ¬ φ ⊨ ⊥ only if Γ ⊨ φ. Now φ is just the lemma we need in order to use the premise ¬ φ to reach the goal ⊥ and complete the *reductio*. The fact that our goal is ⊥ tells us that it is safe to set φ as a goal, and the law for negation as a premise tells us that it is safe to drop ¬ φ from the active resources of the gap in which we establish the lemma φ. That is, we can use a negation ¬ φ to complete any *reductio* argument, so we can exploit a negated compound whenever our goal is ⊥.

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

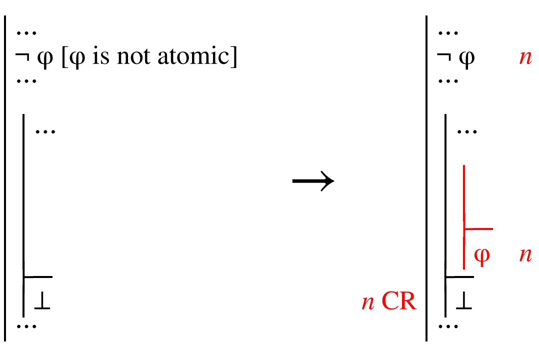


Fig. 3.3.2-1. Developing a derivation by exploiting a negated compound at stage *n*.

The motivation for CR lies in its use to exploit the negations of non-atomic sentences, since we can arrange things so that the negations of atomic sentences remain active forever. In fact, we must limit the use of CR to the negations of non-atomic sentences. It is sound and safe in the case of negations of atomic sentences, but it would not be progressive in that case because it would allow us to go around in circles. Both IP and CR carry us between gaps whose proximate arguments have the forms Γ, ¬ φ / ⊥ and Γ / φ; but they carry us in opposite directions, so, if there is any overlap in the sentences φ to which they apply, a derivation could move back and forth between the two arguments forever. We block such circles by limiting IP to cases where φ is atomic and limiting CR to cases where φ is non-atomic.

One way of understanding the role of CR is to compare it with a use of LFR, where the recourse to lemma is more explicit. Below are two derivations for the argument that was used as an illustration in the last subsection. The one on the left uses CR and the one on the right uses LFR:

	¬ ((A ∧ B) ∧ ¬ C)	2
	A	(3)
	B	(3)
	¬ C	(4)
3 Adj	A ∧ B	X,(4)
4 Adj	(A ∧ B) ∧ ¬ C	X,(5)
	●	
5 QED	(A ∧ B) ∧ ¬ C	2
2 CR	⊥	1
1 IP	C	

	¬ ((A ∧ B) ∧ ¬ C)	(6)
	A	(3)
	B	(3)
	¬ C	(4)
3 Adj	A ∧ B	X,(4)
4 Adj	(A ∧ B) ∧ ¬ C	X,(5)
	●	
5 QED	(A ∧ B) ∧ ¬ C	2
	(A ∧ B) ∧ ¬ C	(6)
	●	
6 Nc	⊥	2
2 LFR	⊥	1
1 IP	C	

Notice that the gap resulting from CR on the left is identical to, and filled in the same way as, the first of the two gaps introduced by LFR on the right. We know in advance that the second of these gaps will close because the denial of its supposition is one or our active resources. Indeed the point of choosing (A ∧ B) ∧ ¬ C as the lemma in LFA is to combine it with the resource ¬ ((A ∧ B) ∧ ¬ C) to reach ⊥ and complete the *reductio*. That is, LFA on the right is part of a plan to use the first premise. What is new in CR is the claim that this resource need not be used further in developing the derivation and may be dropped from its active resources. And this makes CR clearly progressive in a way that LFR is not.

3.3.3. More examples

Here is an English argument whose derivation exhibits all of the rules for negation:

Ann's proposal wasn't unfunded without
Bill's and Carol's each being funded
Bill's proposal was not funded

Ann's proposal was funded

And here is the derivation:

	¬ (¬ A ∧ ¬ (B ∧ C))	2
	¬ B	(7)
	│ ¬ A	(4)
	│ │ │ ●	
4 QED	│ │ │ ¬ 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	

The rules of this section are used at the first two stages, and the rules of 3.2 are in the course of reaching the goal introduced by CR. One alternative approach would be to introduce ¬ (B ∧ C) as a lemma at the second stage using LFR. Combined with a use of Adj to add ¬ A ∧ ¬ (B ∧ C) as a resource, it would produce a simpler derivation but one that requires foresight to discover.

In the absence of the rules of this section, the exercise **2d** of 3.2.x required use of LFR. Here are two derivations for the argument of that exercise which use CR instead but differ in the choice of the premise to be exploited by this rule.

	¬ (A ∧ B)	3
	¬ (C ∧ ¬ B)	(8)
	│ A ∧ C	2
2 Ext	│ A	(5)
2 Ext	│ C	(7)
	│ │ │ ●	
5 QED	│ │ │ A	4
	│ │ │ │ ¬ B	(7)
7 Adj	│ │ │ │ C ∧ ¬ B	X,(8)
	│ │ │ │ ●	
8 Nc	│ │ │ │ ⊥	6
6 IP	│ │ │ B	4
4 Cnj	│ │ A ∧ B	3
3 CR	│ ⊥	1
1 RAA	¬ (A ∧ C)	

	¬ (A ∧ B)	(8)
	¬ (C ∧ ¬ B)	3
	│ A ∧ C	2
2 Ext	│ A	(7)
2 Ext	│ C	(5)
	│ │ │ ●	
5 QED	│ │ │ C	4
	│ │ │ │ B	(7)
7 Adj	│ │ │ │ A ∧ B	X,(8)
	│ │ │ │ ●	
8 Nc	│ │ │ │ ⊥	6
6 RAA	│ │ │ ¬ B	4
4 Cnj	│ │ C ∧ ¬ B	3
3 CR	│ ⊥	1
1 RAA	¬ (A ∧ C)	

These derivations have the same number of stages as the answer in 3.2.xa for **2d**, but their scope lines are nested one deeper. Each of the arguments completing the gaps set up by LFR in the earlier derivation appears in one of these derivations, but we arrive at these arguments in a different way.

It is possible to dispense with Adj in the derivations above and exploit both premises by CR. This leads to a derivation with two more stages and scope lines that are nested more deeply. What we get in return for that increased complexity is direction in how to complete the derivation. In effect, all the thinking required to identify appropriate lemmas is done on paper. We will look at this third approach to the example in 3.5, where we consider how the rules guide the search for derivations.

3.3.s. Summary

1 The law for negation as a premise tells us two things about entailment. It tells us first that a conclusion is valid if and only if the denial of that conclusion can be reduced to absurdity given the premises. This is the principle of indirect proof; it is closely tied to the entailment $\neg \neg \varphi \vDash \varphi$ (and is subject to the same concerns as is that entailment). We have no need for this principle except in the case of unanalyzed components, which we will begin to call atomic sentences. And, for reasons noted later, we need to limit the use of the rule Indirect Proof (IP) to such conclusions.

2 Another lesson we can draw from the law for negation as a premise is that a *reductio* argument with a negative premise $\neg \varphi$ is valid if and only if the sentence φ is entailed by whatever other premises there are. This tells us that φ can be safely introduced as a lemma even if we drop $\neg \varphi$ from our active resources. The rule implementing this idea, Completing a *Reductio* (CR) serves as our rule for exploiting negative resources. It applies only to *reductio* arguments but the availability of IP insures that any gap will eventually turn into a gap in a *reductio* argument (unless it closes before that point). Since CR, by dropping a resource $\neg \varphi$ and adding a goal φ has an effect opposite to that of IP, we must apply them to different sentences φ to avoid going in circles. So, just as IP is limited to atomic sentences, CR is limited to negations of non-atomic sentences.

3 The rule CR can lead us to set as goals any lemmas we need in order to use negations in completing *reductio* arguments. It therefore eliminates any need for LFR. The rule Adj is also no longer needed (though still sometimes useful) since the rules CR and Cnj will lead us to identify and prove any lemma that Adj would introduce. Indeed, derivations for arguments involving conjunction can now be constructed by simply letting the rules guide us.

Glen Helman 03 Aug 2010

3.3.x. Exercise questions

Use derivations to establish each of the claims of entailment shown below. You can maximize your practice in the use of CR by avoiding LFR and using Adj only in cases where the goal is a conjunction.

1. $\neg (A \land \neg B), A \vDash B$

2. $J \land \neg (J \land \neg C) \vDash J \land C$ (see exercise **1j** of 3.1.x)

3. $\neg (\neg (A \land B) \land C), \neg A \vDash \neg C$

4. $\neg (A \land \neg (B \land C)) \vDash \neg (A \land \neg B)$

5. $\neg (A \land \neg B), \neg (B \land \neg C) \vDash \neg (A \land \neg C)$

6. $\neg (A \land \neg B), \neg (A \land \neg C) \vDash \neg (A \land \neg (B \land C))$

For more exercises, use the exercise machine.

Glen Helman 03 Aug 2010

3.3.xa. Exercise answers

1.

	$\neg (A \land \neg B)$	2
	A	(3)
	│ $\neg B$	(3)
3 Adj	│ │ $A \land \neg B$	X,(4)
	│ │ ●	
4 QED	│ │ $A \land \neg B$	2
2 CR	│ $\bot$	1
1 IP	B	

2.

	$J \land \neg (J \land \neg C)$	1
1 Ext	J	(3),(6)
1 Ext	$\neg (J \land \neg C)$	5
	│ ●	
3 QED	│ J	2
	│ │ $\neg C$	(6)
6 Adj	│ │ │ $J \land \neg C$	X,(7)
	│ │ │ ●	
7 QED	│ │ │ $J \land \neg C$	5
5 CR	│ │ $\bot$	4
4 IP	│ C	2
2 Cnj	$J \land C$	

3.

	$\neg (\neg (A \land B) \land C)$	2
	$\neg A$	(7)
	│ C	(4)
	│ │ │ $A \land B$	6
6 Ext	│ │ │ A	(7)
6 Ext	│ │ │ B	
	│ │ │ ●	
7 Nc	│ │ │ $\bot$	5
5 RAA	│ │ $\neg (A \land B)$	3
	│ │ ●	
4 QED	│ │ C	3
3 Cnj	│ │ $\neg (A \land B) \land C$	2
2 CR	│ $\bot$	1
1 RAA	$\neg C$	

4.

	$\neg (A \land \neg (B \land C))$	3
	│ $A \land \neg B$	2
2 Ext	│ A	(5)
2 Ext	│ $\neg B$	(8)
	│ │ ●	
5 QED	│ │ A	4
	│ │ │ $B \land C$	7
7 Ext	│ │ │ B	(8)
7 Ext	│ │ │ C	
	│ │ │ ●	
8 Nc	│ │ │ $\bot$	6
6 RAA	│ │ $\neg (B \land C)$	4
4 Cnj	│ │ $A \land \neg (B \land C)$	3
3 CR	│ $\bot$	1
1 RAA	$\neg (A \land \neg B)$	

5.

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

6.

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

Choosing ¬ (B ∧ C) as the resource to exploit by CR at stage 3 would lead to a somewhat shorter and simpler derivation.

Glen Helman 03 Aug 2010

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.

3.4.1. When *reductio*s fail
Changes in the arguments used to show the sufficiency, conservativeness, and decisiveness of the system of derivations correspond to changes in the way we present counterexamples.

3.4.2. Some examples of consistency
When a *reductio* fails, we know that its premises are not inconsistent, so derivations that fail will now lead us to consistent sets of sentences.

Glen Helman 03 Aug 2010

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

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

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.

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.

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. ¬ B / ¬ (A ∧ ¬ B)

b. ¬ (A ∧ B) / ¬ A ∧ ¬ B

c. ¬ (A ∧ B), ¬ (B ∧ C) / ¬ (A ∧ 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. ¬ (A ∧ ¬ B) ⊨ B

b. ¬ (A ∧ B) ⊨ ¬ (B ∧ A)

c. ¬ (A ∧ ¬ B) ⊨ ¬ (B ∧ ¬ A)

d. ¬ (A ∧ B), ¬ (B ∧ C), B ⊨ ¬ A ∧ ¬ C

e. ¬ (A ∧ ¬ (B ∧ ¬ (C ∧ ¬ D))) ⊨ ¬ (A ∧ ¬ (B ∧ D))

For more exercises, use the exercise machine.

Glen Helman 03 Aug 2010

3.4.xa. Exercise answers

1. a.

	│ ¬ B	
	│ │ A ∧ ¬ B	2
2 Ext	│ │ A	
2 Ext	│ │ ¬ B	
	│ │ ○	A, ¬ B ⊭ ⊥
	│ │ ⊥	1
1 RAA	│ ¬ (A ∧ ¬ B)	

A	B	¬ B / ¬ (A ∧ ¬ B)
T	F	Ⓣ Ⓕ T T

b.

	│ ¬ (A ∧ B)	3,8
	│ │ │ A	(5)
	│ │ │ │ │ ●	
5 QED	│ │ │ │ │ A	4
	│ │ │ │ │ │ ¬ B	
	│ │ │ │ │ │ ○	A, ¬ B ⊭ ⊥
	│ │ │ │ │ │ ⊥	6
6 IP	│ │ │ │ │ B	4
4 Cnj	│ │ │ │ A ∧ B	3
3 CR	│ │ │ ⊥	2
2 RAA	│ │ ¬ A	1
	│ │ │ B	(11)
	│ │ │ │ │ │ ¬ A	
	│ │ │ │ │ │ ○	¬ A, B ⊭ ⊥
	│ │ │ │ │ │ ⊥	10
10 IP	│ │ │ │ │ A	9
	│ │ │ │ │ ●	
11 QED	│ │ │ │ │ B	9
9 Cnj	│ │ │ │ A ∧ B	8
8 CR	│ │ │ ⊥	7
7 RAA	│ │ ¬ B	1
1 Cnj	│ ¬ A ∧ ¬ B	

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

The first row is an interpretation that divides the first gap; and the second row is an interpretation that 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.

	$\neg (A \land B)$	3,8
	$\neg (B \land C)$	11
	B	(6),(10),(14)
	A	(5)
	●	
5 QED	A	4
	●	
6 QED	B	4
4 Cnj	$A \land B$	3
3 CR	$\bot$	2
2 RAA	$\neg A$	1
	C	(15)
	$\neg A$	
	●	
14 QED	B	13
	●	
15 QED	C	13
13 Cnj	$B \land C$	12
12 CR	$\bot$	11
11 IP	A	9
	●	
10 QED	B	9
9 Cnj	$A \land B$	8
8 CR	$\bot$	7
7 RAA	$\neg C$	1
1 Cnj	$\neg A \land \neg C$	

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

The derivation on the left exploits resources in their order of appearance; while the one above 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 on the left is eventually led to exploit the same resource to the same effect.

e.

	$\neg (A \land \neg (B \land \neg (C \land \neg D)))$	3
	$A \land \neg (B \land D)$	2
2 Ext	A	(5)
2 Ext	$\neg (B \land D)$	8
	●	
5 QED	A	4
	$B \land \neg (C \land \neg D)$	7
7 Ext	B	(10)
7 Ext	$\neg (C \land \neg D)$	12
	●	
10 QED	B	9
	$\neg D$	(15)
	$\neg C$	
	○	$A, B, \neg C, \neg D \nvDash \bot$
	$\bot$	14
14 IP	C	13
	●	
15 QED	$\neg D$	13
13	$C \land \neg D$	12
12 CR	$\bot$	11
11 IP	D	9
9 Cnj	$B \land D$	8
8 CR	$\bot$	6
6 RAA	$\neg (B \land \neg (C \land \neg D))$	4
4 Cnj	$A \land \neg (B \land \neg (C \land \neg D))$	3
3 CR	$\bot$	1
1 RAA	$\neg (A \land \neg (B \land D))$	

A B C D	$\neg (A \land \neg (B \land \neg (C \land \neg D)))$ / $\neg (A \land \neg (B \land D))$
T T F F	Ⓣ F F T T F T Ⓕ T T F

Glen Helman 03 Aug 2010

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 03 Aug 2010

3.5.1. Approaching derivations

If we set aside the rules LFR and Adj, the general advice 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 rules other than LFR and Adj 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 whenever 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 further rules LFR and Adj 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 above are labeled *basic rules* and are counted as part of the *basic system of derivations*.

Glen Helman 03 Aug 2010

3.5.2. An example

As an example of the use of the basic system, let us look at the further derivation for the argument of 3.2.x **2d** that was promised in 3.3.3. 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.

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

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.

	$\neg (A \wedge B)$	
	$\neg (C \wedge \neg B)$	
	$A \wedge C$	
	$\bot$	1
1 RAA	$\neg (A \wedge C)$	

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.

	$\neg (A \wedge B)$	
	$\neg (C \wedge \neg B)$	
	$A \wedge C$	2
2 Ext	A	
2 Ext	C	
	$\bot$	1
1 RAA	$\neg (A \wedge C)$	

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 ⊥.

	$\neg (A \wedge B)$	3
	$\neg (C \wedge \neg B)$	
	$A \wedge C$	2
2 Ext	A	
2 Ext	C	
	$A \wedge B$	3
3 CR	$\bot$	1
1 RAA	$\neg (A \wedge C)$	

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.)

	$\neg (A \wedge B)$	3
	$\neg (C \wedge \neg B)$	
	$A \wedge C$	2
2 Ext	A	
2 Ext	C	
	A	4
	B	4
4 Cnj	$A \wedge B$	3
3 CR	$\bot$	1
1 RAA	$\neg (A \wedge C)$	

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.

```
         │ ¬ (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. 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.

```
         │ ¬ (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. 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.

```
         │ ¬ (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. 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.

	$\neg (A \land B)$	3
	$\neg (C \land \neg B)$	7
	$A \land C$	2
2 Ext	A	(5)
2 Ext	C	
	●	
5 QED	A	4
	$\neg B$	
	C	8
	$\neg B$	8
8 Cnj	$C \land \neg B$	7
7 CR	$\bot$	6
6 IP	B	4
4 Cnj	$A \land B$	3
3 CR	$\bot$	1
1 RAA	$\neg (A \land C)$	

The complete derivation is shown below.

	$\neg (A \land B)$	3
	$\neg (C \land \neg B)$	7
	$A \land C$	2
2 Ext	A	(5)
2 Ext	C	(9)
	●	
5 QED	A	4
	$\neg B$	(10)
	●	
9 QED	C	8
	●	
10 QED	$\neg B$	8
8 Cnj	$C \land \neg B$	7
7 CR	$\bot$	6
6 IP	B	4
4 Cnj	$A \land B$	3
3 CR	$\bot$	1
1 RAA	$\neg (A \land C)$	

Glen Helman 03 Aug 2010

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 is 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 ⊤, ⊥, 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:

   Rules for closing gaps

rule	*conditions for applying it*
QED	the goal is among the resources
Nc	the goal is ⊥, and there are sentences φ and ¬ φ among the resources
ENV	the goal is ⊤
EFQ	⊥ 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.

   Exploitation and planning rules

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 ⊥)	RAA
negated	atomic sentence	*none*	RAA
	atomic sentence	*none*	IP
	⊤ or ⊥	*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 on 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 when you see that they are possible. Such shortcuts include the use of available but inactive resources with rules like QED and the use of non-basic rules like Adj and LFR.

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 those rules 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
sentences φ and $\neg\varphi$ are resources & the goal is $\bot$	Nc
$\top$ is the goal	ENV
$\bot$ is a resource	EFQ

Basic system

Added rules (optional)

Attachment rule: *added resource*	*rule*
$\varphi \wedge \psi$	Adj

Rule for lemmas: *prerequisite*	*rule*
the goal is $\bot$	LFR

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

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

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.

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.

	$\neg A$	(3)
	$B \wedge A$	2
2 Ext	B	
2 Ext	A	(3)
	●	
3 Nc	$\bot$	1
1 RAA	$\neg (B \wedge A)$	

1. One open gap. $\neg A / \neg (B \wedge A)_{[RAA]}$. Developled.
2. One open gap. $\neg A, B \wedge A_{[Ext]} / \bot$. Developed.
3. One open gap. $\neg A, B, A /_{[Nc]} \bot$. Closed gap.
4. No gaps open.

b.

	$A \wedge B$	1
1 Ext	A	(3)
1 Ext	B	(4)
	●	
3 QED	B	2
	●	
4 QED	A	2
2 Cnj	$B \wedge A$	

1. One open gap. $A \wedge B_{[Ext]} / B \wedge A_{[Cnj]}$. Developed by Ext.
2. One open gap. $A, B / B \wedge A_{[Cnj]}$. Developed.
3. Chose first open gap. $A, B /_{[QED]} A$. Closed gap.
4. One open gap. $A, B /_{[QED]} B$. Closed gap.
5. No gaps open.

c.

	B	(2)
	●	
2 QED	B	1
	$\neg A$	
	○	$B, \neg A \nvDash \bot$
	$\bot$	3
3 IP	A	1
1 Cnj	$B \wedge A$	

1. One open gap. $B / B \wedge A_{[Cnj]}$. Developed.
2. First open gap. $B /_{[QED]} B$. Closed gap.
3. One open gap. $B / A_{[IP]}$. Developed.
4. One open gap. $B, \neg A / \bot$. Dead end.

d.

	$\neg (A \wedge B)$	2
	A	(4)
	B	(5)
	●	
4 QED	A	3
	●	
5 QED	B	3
3 Cnj	$A \wedge B$	2
2 CR	$\bot$	1
1 RAA	$\neg B$	

1. One open gap. $\neg (A \wedge B), A / \neg B_{[RAA]}$. Developed.
2. One open gap. $\neg (A \wedge B)_{[CR]}, A, B / \bot$. Developed.
3. One open gap. $A, B / A \wedge B_{[Cnj]}$. Developed.
4. Chose first open gap. $A, B /_{[QED]} A$. Closed gap.
5. One open gap. $A, B /_{[QED]} B$. Closed gap.
6. No gaps open.

e.

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

1. One open gap. $\neg (A \wedge B), \neg (B \wedge C) / \neg B_{[RAA]}$. Developed.
2. One open gap. $\neg (A \wedge B)_{[CR]}, \neg (B \wedge C)_{[CR]}, B / \bot$. Developed by first CR.
3. One open gap. $\neg (B \wedge C), B / A \wedge B_{[Cnj]}$. Developed.
4. Chose second open gap. $\neg (B \wedge C), B /_{[QED]} B$. Closed gap.
5. One open gap. $\neg (B \wedge C), B / A_{[IP]}$. Developed.
6. One open gap. $\neg (B \wedge C)_{[CR]}, B, \neg A / \bot$. Developed.
7. One open gap. $B, \neg A / B \wedge C_{[Cnj]}$. Developed.
8. Chose first open gap. $B, \neg A /_{[QED]} B$. Closed gap.
9. One open gap. $B, \neg A / C_{[IP]}$. Developed.
10. One open gap. $B, \neg A, \neg C / \bot$. 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 1st premise instead of 2nd
† 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 2nd premise instead of 1st
† 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 03 Aug 2010