3.2. *Reductio* arguments: refuting suppositions

3.2.0. Overview

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

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

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 16 Sep 2004

3.2.1. The duality of premises and alternatives

The basic law for relative exhaustiveness tells us that, when sentences ϕ and ϕ' are contradictory, having one as a premise comes to the same thing as having the other as a conclusion—that is,

$$\Gamma \Rightarrow \phi, \Delta \text{ if and only if } \Gamma, \phi' \Rightarrow \Delta$$

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

$$\Gamma \Rightarrow \neg\phi, \Delta \text{ if and only if } \Gamma, \phi \Rightarrow \Delta$$
$$\Gamma, \neg\phi \Rightarrow \Delta \text{ if and only if } \Gamma \Rightarrow \phi, \Delta$$

where we get the second by turning around both the pair of contradictories and the *if-and-only-if* claim. That is, having a negation as either a premise or alternative comes to the same thing as having the unnegated sentence in the opposite role. The kind opposition in question here is the sort of duality mentioned in 1.4.6.

We do not study relative exhaustiveness directly, and one use of the basic law for relative exhaustiveness is to exchange alternatives for premises so that a claim of relative exhaustiveness may be converted into a claim of entailment. But suppose we begin with only a single alternative; that is, suppose Δ is empty. In this case, when ϕ and ϕ' are contradictory, we can say that

$$\Gamma \Rightarrow \phi \text{ if and only if } \Gamma, \phi' \Rightarrow$$

where the right-hand side says that ϕ' is inconsistent with (or is excluded by) Γ. When we express inconsistency by way of the validity of a *reductio* argument, we get the following ***basic law for contradictories***:

$$\text{if } \phi \text{ and } \phi' \text{ are contradictory, then } \Gamma \Rightarrow \phi \text{ if and only if } \Gamma, \phi' \Rightarrow \bot$$

Indeed, if the right-hand side of this holds for every choice of the set Γ, then ϕ and ϕ' must be contradictory. (To see why, think what follows if Γ is chosen first as the single sentence ϕ and then as the sentence $\neg\phi'$.)

Now we can get some basic principles for negation with regard to entailment by applying the basic law for contradictories to the case of negation in the way we applied the basic law for relative exhaustiveness above. That is, if we take ϕ and $\neg\phi$ as our contradictory sentences, we get:

Law for negation as a conclusion. $\Gamma \Rightarrow \neg\, \phi$ if and only if $\Gamma, \phi \Rightarrow \bot$
Law for negation as a premise. $\Gamma, \neg\, \phi \Rightarrow \bot$ if and only if $\Gamma \Rightarrow \phi$

again reversing the pair of contradictories and the *if-and-only-if* claim to get the second.

Although these principles are dual in something like the way that the earlier pair for relative 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 $\neg\, \neg\, \phi$ 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 that we will postpone until §3.3.

Glen Helman 16 Sep 2004

3.2.2. Drawing negative conclusions

The law for negation as a conclusion

$\Gamma \Rightarrow \neg\, \phi$ if and only if $\Gamma, \phi \Rightarrow \bot$

describes the conditions under which an entailment of the form $\Gamma \Rightarrow \neg\, \phi$ 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. The negative conclusion of this argument is valid because the conclusion denies something that is excluded by the premises.

And the law for negation as a conclusion says that this holds in general since a *reductio* entailment $\Gamma, \phi \Rightarrow \bot$ is the way we capture exclusion in terms of entailment. That is, Γ excludes ϕ if adding ϕ to Γ would enable us to reach an absurd conclusion. In these terms, we can validly conclude a negation $\neg\, \phi$ when we can reduce to absurdity the result of adding ϕ to the premises Γ.

Although the entailment $\Gamma, \phi \Rightarrow \bot$ shows the inconsistency of the full set containing the members of Γ together with ϕ, we focus attention on ϕ when we say it is excluded by, or is inconsistent with, Γ. Similarly, we can say that the argument $\Gamma, \phi \,/\, \bot$ reduces ϕ to absurdity given Γ. So we can restate the law for negation as a conclusion in another way: we can validly conclude a negation $\neg\, \phi$ from premises Γ when we can reduce ϕ to absurdity given the premises Γ.

To implement this idea as a way of developing a gap whose goal is $\neg\, \phi$, 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 second gap introduced by the rules Lemma and LFR of §2.4, the lemma is introduced as a supposition. In those rules it represents a resource that we have on loan, a loan that is paid if we are able to close the first 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 the assumptions to which the supposition was added. 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 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. That is, the rule by which we plan for negative goals will take the form shown in Figure 3.2.2-1.

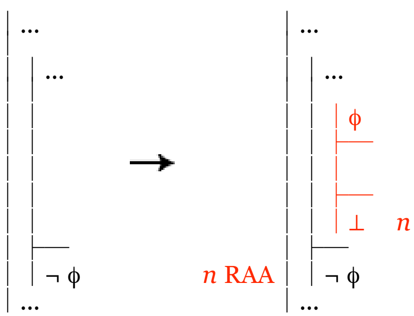
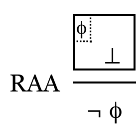


Fig. 3.2.2-1. Developing a derivation by planning for a negation at stage *n*.

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):



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 assumption. The fact that ⊥ falls within the scope of a supposition ϕ in the tree-form argument, shows that it is itself the conclusion of a *reductio* argument. The rule RAA enables us to transform that argument by both weakening its conclusion and ending the scope of the assumption ϕ.

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***. Even though this is sometimes referred to as the "law of contradiction" and its name is more consistent than the way it is stated, the basic idea is always the same: a statement and its denial cannot both be true. We know it as the principle that ¬ ϕ and ϕ are mutually exclusive—or, in the form most relevant at the moment, that ¬ ϕ, ϕ ⇒ ⊥. This idea lies behind a pattern of argument that we will call ***Non-contradiction*** or Nc:

$$\text{NC}\ \frac{\neg\,\phi \quad \phi}{\bot}$$

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

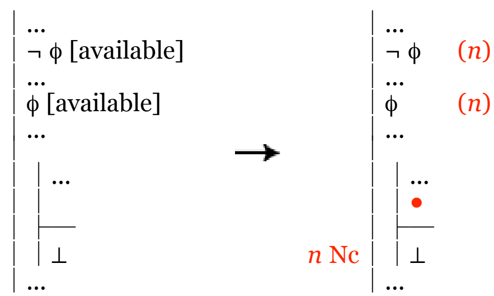


Fig. 3.2.2-2. Closing the gap of a *reductio* argument one of whose resources negates another.

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, of course, moot since the gap closes and, in a way, the first point 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 01 Aug 2004

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 possible to have all uses of Ext at the beginning of the derivation since some of them are used to exploit the supposition B ∧ (C ∧ ¬ D)). Of course, we might have used Reductio Ad Absurdum and made this supposition at the first stage and then applied Ext to all the resources that had accumulated. But the following derivation shows that even this degree of grouping will not always be possible.

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

We might have waited until after the supposition B ∧ C was made before applying Ext 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.

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*)

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

`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:

$$\frac{\neg ((A \land B) \land \neg C) \quad A \land \neg C}{\neg B}$$

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., $(A \land B) \land \neg C$—as a resource, we need to use Adjunction to add its first conjunct and the full compound.

	│ $\neg ((A \land B) \land \neg C)$	(6)
	│ $A \land \neg C$	2
	├──	
2 Ext	│ A	(4)
2 Ext	│ $\neg C$	(5)
	│ │ B	(4)
	│ ├──	
4 Adj	│ │ $A \land B$	X,(5)
5 Adj	│ │ $(A \land B) \land \neg C$	X,(6)
	│ │ •	
	│ ├──	
6 NC	│ │ ⊥	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	│ │ ⊥	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 the derivation. This is not the sort of entailment where such an explicit argument would ordinarily be given, but 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.

3.2.s. Summary

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, the basic law for contradictories, 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.

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.

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 direct rule introduced in the next section.

Glen Helman 01 Aug 2004

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.** ¬ A ⇒ ¬ (A ∧ B)
- **b.** ¬ B ⇒ ¬ (A ∧ B) ∧ ¬ (B ∧ C)
- **c.** ⇒ ¬ (A ∧ ¬ A)
- **d.** J ∧ C ⇒ J ∧ ¬ (J ∧ ¬ 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.** ¬ (A ∧ B), A ⇒ ¬ B
- **b.** ¬ (A ∧ ¬ B), ¬ B ⇒ ¬ A
- **c.** A, ¬ (A ∧ B), ¬ (A ∧ C) ⇒ ¬ B ∧ ¬ C
- **d.** ¬ (A ∧ B), ¬ (C ∧ ¬ B) ⇒ ¬ (A ∧ C)

Glen Helman 16 Sep 2004

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)	