# 2.2. Proofs: analyzing entailment

## 2.2.0. Overview

Some insight into deductive logic can by looking at basic principles of entailment, but more will come by looking at how these principles may be combined in proofs.

2.2.1. Proofs as trees
: The simplest way of combining deductive principles takes the shape of a tree in which premises, premises for premises, and so on, grow and branch from the final conclusion.

2.2.2. Derivations
: Although tree-form notation can make the structure of a proof very explicit, we will mainly use a compact notation that more closely matches the patterns that are used when deductive reasoning is put into words.

2.2.3. Rules for derivations
: In the context of derivations, principles of entailment take the form of rules that direct the search for a proof.

2.2.4. An example
: All derivations involving conjunction alone share many features; we will look closely at one typical example.

2.2.5. More rules
: Tautology and absurdity provide the first example of derivation rules for logical forms other than conjunction.

2.2.6. Resources
: In order to plot a course in constructing a proof for a given conclusion, we need to keep track of not only the premises but also the conclusions that have already been reached.

Glen Helman 15 Aug 2006

## 2.2.1. Proofs as trees

Our study of entailments involving conjunction will rest on the principles discussed in 2.1.1. These are shown below in symbolic form on the left and in English on the right:

| | |
|---|---|
| $\varphi \wedge \psi \Rightarrow \varphi$ | both $\varphi$ and $\psi \Rightarrow \varphi$ |
| $\varphi \wedge \psi \Rightarrow \psi$ | both $\varphi$ and $\psi \Rightarrow \psi$ |
| $\varphi, \psi \Rightarrow \varphi \wedge \psi$ | $\varphi, \psi \Rightarrow$ both $\varphi$ and $\psi$. |

We will refer to the first two of these patterns as ***extraction*** (***left*** and ***right*** extraction when we wish to distinguish them) and to the third simply as ***conjunction***. To establish particular cases of entailment, we will want to link together special cases of these general patterns and, eventually, of other patterns, too.

One notation for doing that employs something like the two-dimensional form we have used for arguments, with the conclusion below the premises and marked off from them by a horizontal line. In order to make the premises of a multi-premised argument available to serve as conclusions of further argument, we will spread them out horizontally. In this style of notation, the basic patterns for conjunction take the following forms (where abbreviations of their names are used as labels):

$$\text{Ext}\ \frac{\varphi \wedge \psi}{\varphi} \qquad \text{Ext}\ \frac{\varphi \wedge \psi}{\psi} \qquad \text{Cnj}\ \frac{\varphi \quad \psi}{\varphi \wedge \psi}$$

Arguments exhibiting these patterns can be linked by treating the premises of one argument as conclusions of other arguments. For example, the following shows that $(A \wedge B) \wedge C$ is a valid conclusion from the two premises $A$ and $B \wedge C$:

$$\text{Cnj}\ \dfrac{\text{Cnj}\ \dfrac{A \qquad \text{Ext}\ \dfrac{B \wedge C}{B}}{A \wedge B} \qquad \text{Ext}\ \dfrac{B \wedge C}{C}}{(A \wedge B) \wedge C}$$

The ability to put the principles for conjunction together in this way rests on the general laws of entailment discussed in 1.4.7. The law for premises enables us to begin; it shows that the premises $A$ and $B \wedge C$ entail the tips of the branches of this tree-like proof. Repeated uses of the chain law then enable to add conclusions drawn using the principles for conjunction, and we work our way down the tree showing that the original set of premises entails each intermediate conclusion and, eventually, $(A \wedge B) \wedge C$. For example, just before the end, we know that our original premises entail each of the premises of the final conclusion—i.e., that $A, B \wedge C \Rightarrow A \wedge B$ and $A, B \wedge C \Rightarrow C$. The chain law then enables us to combine these entailments with the fact that $A \wedge B, C \Rightarrow (A \wedge B) \wedge C$ (a case of Conjunction) to show that $A, B \wedge C \Rightarrow (A \wedge B) \wedge C$.

These tree-form proofs are less helpful in the case of other connectives; and, in 2.3, we will look at a different way of arguing in the case of conjunction that is better suited to other logical forms. We can make one step in that direction now by looking at some basic principles for entailment that describe the conditions under which any arguments involving conjunction are valid.

Law for conjunction as a premise. $\Gamma, \varphi \wedge \psi \Rightarrow \chi$ if and only if $\Gamma, \varphi, \psi \Rightarrow \chi$

Law for conjunction as a conclusion. $\Gamma \Rightarrow \varphi \wedge \psi$ if and only if both $\Gamma \Rightarrow \varphi$ and $\Gamma \Rightarrow \psi$

These principles can be seen to hold by the comparing the sort of possible worlds each side of the *if and only if* rules out. In the first principles, each side rules out the possibility of a world in which $\chi$ is false while $\varphi$, $\psi$, and the members of $\Gamma$ are all true; that means that these two entailments offer equivalent guarantees, so each holds if and only if the other does. In the second principle, the sort of worlds ruled out by guarantee on the left are the worlds in which the members of $\Gamma$ are all true but $\varphi$ or $\psi$ is false, and the same worlds are ruled out when we have both the guarantees on the right. The upshot is that these two principles suffice, together with the law of premises, to establish any cases of validity that depend on conjunction alone.

The *if* part of these principles reflects the validity of arguments of the forms Ext and Cnj (together with the chain law). The *only if* part of the first tells us that whatever a conjunction contributes as a premise of a valid argument is already contributed by the conclusions we could derive by Ext; that is, our use of a conjunction among the premises need only be by way of Ext. The *only if* part of the second tells us that, if a conjunction is a valid conclusion, then the premises needed to reach it by Cnj are themselves valid conclusions. When conjunction is the only connective employed in our analysis of sentences, applying these two principles repeatedly will eventually bring us back to arguments whose premises and conclusions are all unanalyzed components. If these components are logically independent, an argument whose premises and conclusion are drawn from them is valid when and only when its conclusion is among its premises; thus, if it is valid, its validity follows by the law of premises.

A couple of the principles for $\top$ and $\bot$—those for $\top$ as a conclusion and $\bot$ as a premise—assert the validity of arguments and can be used to build tree-form proofs:

$$\text{ENV}\ \frac{\quad}{\top} \qquad \text{EFQ}\ \frac{\bot}{\varphi}$$

The label for the second, EFQ, abbreviates the Latin *ex falso quodlibet* (which might be translated as *from the false, whatever*), a traditional description the law for $\bot$ as a premise, and the label for the first, ENV, abbreviates *ex nihilo verum* (*from nothing, the true*), which gives a corresponding description of the law for $\top$ as a conclusion.

The argument ENV has no premises and serves to close off a branch of a tree, making it one that need not have a premise at its tip—as in the following proof, which shows that $A, B \Rightarrow (B \wedge \top) \wedge A$:

$$\text{Cnj}\ \dfrac{\text{Cnj}\ \dfrac{B \qquad \text{ENV}\ \dfrac{\quad}{\top}}{B \wedge \top} \qquad A}{(B \wedge \top) \wedge A}$$

The pattern EFQ enables us to connect a proof ending with the conclucsion $\bot$ to the tip of any branch. The following example uses it to show that $A \wedge (\bot \wedge B) \Rightarrow C \wedge D$:

$$\text{Cnj}\ \dfrac{\text{EFQ}\ \dfrac{\text{Ext}\ \dfrac{\text{Ext}\ \dfrac{A \wedge (\bot \wedge B)}{\bot \wedge B}}{\bot}}{C} \qquad \text{EFQ}\ \dfrac{\text{Ext}\ \dfrac{\text{Ext}\ \dfrac{A \wedge (\bot \wedge B)}{\bot \wedge B}}{\bot}}{D}}{C \wedge D}$$

The premise $A \wedge (\bot \wedge B)$ is the starting point for proofs ending in A and B, too, but these proofs would never be needed. Even if A and B were required as premises in an argument, the proof ending with $\bot$ would have been enough to yield them just as it yields C and D.

The two other laws for ⊤ and ⊥ have a different significance. The law for ⊤ as a premise does not correspond to any pattern of valid argument. It merely tells us that any proof ending with ⊤ would contribute nothing to a larger proof and may be ignored. Of course, such a proof might be connected to a branch of proof that has ⊤ at its tip; but a branch like that could be closed off by ENV instead. The law for ⊥ as an alternative does not figure as a principle governing the construction of proofs at all; recall that we have no law for ⊥ as a *conclusion*. Having ⊥ as a conclusion marks a proof as a proof of inconsistency, and such proofs are constructed solely by drawing conclusions from their premises.

Glen Helman 15 Aug 2006

### 2.2.2. Derivations

The tree-form proofs of the last section are probably the clearest way of presenting the structure of proofs; however, they are not very compact. This is in part because they are two-dimensional and in part because the premises and any conclusions reached from them by Ext may be repeated several times in the proof. In practice, we will use a more linear, though still somewhat two-dimensional, notation. We will gain compactness by writing premises and conclusions in a more-or-less vertical way and by minimizing the repetition of premises that are used draw a number of conclusions. But we also make other gains. The notation is designed to incorporate more directly the process of proof discovery, and the notation will approximate some of the ways the structure of proofs is reflected in the essentially linear way argumentation is presented in language. Indeed, although we will not approach in this way, the notation for proofs could be thought of as a notation for analyzing the form of proofs presented in English that is in some respects analogous to our symbolic notation for analyzing the logical forms of sentences.

This machinery will be, as are tree-form proofs, much more than we need to settle questions of entailment involving only conjunction. But we will need more complex approaches eventually; and, because we have simpler ways of seeing that entailments hold in the case of conjunction, it will be easier to see how and why this method works if we develop it now.

The system to be developed here falls into a broad class often referred to as ***natural deduction systems*** because they replicate, to some extent, natural patterns of reasoning. Such systems were first set out in full in the 1930s by G. Gentzen and also by S. Jaskowski, but some of the key ideas can be found already in the Stoic philosopher Chrysippus (who lived in the 3rd century B. C.). The notation we will be using is an adaptation of notation introduced by F. B. Fitch but our approach to these systems will be influenced heavily by the “semantic tableaux” of E. Beth. (Their ideas are now about 50 years old.)

This system, which we will call a ***system of derivations***, will employ a perspective on proofs that we adopted in the last section whenever we considered ways of restating claims of entailment. If we ask whether an entailment holds, we find ourselves faced with the task of reaching the conclusion from the premises (or showing that it cannot be reached). Let us think of the conclusion as our ***goal*** and of the premises as the ***resources*** we have available in trying to reach that goal. Until we reach the goal, it is separated from our resources by a ***gap*** that it is our aim to close.

We will approach the problem of closing this gap (or showing that it cannot be closed) step by step, at each step analyzing the way our goal may be reached or exploiting our resources by drawing conclusions from them. In making a step of either sort, we will restate our problem with different goals or resources; and we will say that, in restating it in this way, we are ***developing*** the derivation. The problem of closing a gap as seen from this perspective corresponds to the problem of matching needed premises and available conclusions, and the development of a derivation amounts to the process of working forward from premises and backward from conclusions in hopes of making this connection.

We begin with a single gap between the premises and conclusion our initial question concerns. This gap can be closed immediately if the conclusion is among the premises. Otherwise we will analyze our goal (at first the original conclusion) or exploit our resources (at first the original premises). When we analyze a goal, we establish subgoals, intermediate conclusions from which the goal can be reached by Cnj. This divides the problem into subproblems, each focused on the gap between our resources and one of these new goals. As we exploit resources by Exp, we make new resources available to connect to our goals. If the goals and unexploited resources all end up as unanalyzed components, we have done as much as we can to prepare gaps to be closed and, if they cannot be closed at this point, we know that the initial argument is not valid.

A derivation will be written as a more or less vertical list of sentences marked up in various ways to indicate the structure of the corresponding tree-like proof. The subgoals that would be from which a conclusion is reached by Cnj will be written one above the other, each preceded by space for further growth, and chains of conclusions from premises will be collapsed by superimposing any shared segments. In effect, we will construct a tree-form proof disassembled in order to be packed flat, and we will be careful to number the parts in order to indicate how it should be assembled.

We begin in the state shown in Figure 2.2.2-1.

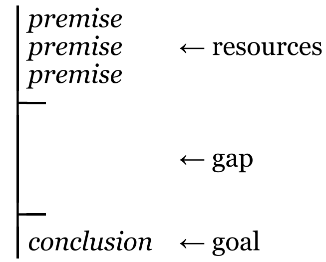



Fig. 2.2.2-1. The initial state of a derivation.

The premises of the argument (if it has any) are written above a horizontal line, and the conclusion is written below a second line. The space in between the horizontal lines marks the gap and will be filled in with additional resources and new goals as the derivation develops.

The vertical line at the left in Figure 2.2.2-1 is a ***scope line*** and will serve us in a number of ways. First of all, new scope lines will be introduced as we analyze goals with a separate scope line serving to mark the portion of the derivation devoted to further analysis of each subgoal. This part of subderivation is where the subgoal is the conclusion we seek to establish, and it is in this sense the scope of the subgoal. As scope lines accumulate, they will be nested, some to the right of others, in a way that indicates the branching of a tree-form proof. In later chapters, proofs will sometimes involve assumptions beyond the initial premises, and scope lines will then also serve to mark the portions of a proof in which these assumptions are being made. Later still, the scope lines will be labeled to indicate vocabulary that has a special role in a portion of a derivation.

At any stage in the development of a derivation, each gap will have certain ***active resources***. These are resources available for use in the gap that have not already been exploited in developing it. Our aim will always be to see whether the goal of a gap is entailed by its active resources.

Glen Helman 15 Aug 2006

## 2.2.3. Rules for derivations

One way of developing a gap is to restate our problem so that one of its resources can be dropped from consideration, perhaps adding others of equivalent power but simpler form. We will call this process ***exploitation***, and it will correspond to a particular way of growing chains of conclusions: always drawing conclusions by both left and right Ext from the end of one of them. The law for conjunction as a premise tells us that anything we can conclude from premises that include a conjunction can still be concluded if we replace the conjunction by its two components. This means that, when we use left and right extraction together, we can eliminate any further need to consider the conjunction we are exploiting. Because we always add both components of a conjunction we exploit, a derivation may contain conclusions that are never used later and would not appear in the corresponding proof tree.

Although we will apply left and right extraction together, we do not duplicate the shared segment of the two resulting chains, so the derivation rule Extraction takes the form shown in Figure 2.2.3-1.

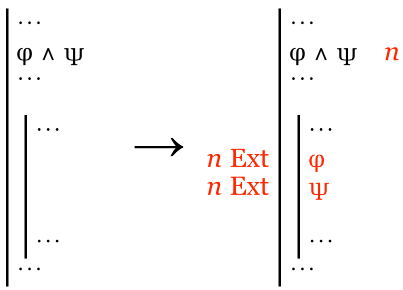



Fig. 2.2.3-1. Developing a derivation by exploiting a conjunction at stage *n*.

On the left, the gap is shown nested inside scope lines (two are shown but there may be just one or more than two). A conjunction is displayed at the top to show that one is among resources available for use in this gap. It is shown to the right of one of the scope lines running to the left of the gap but not the other. The requirement this illustrates is that a resource being exploited need not be inside all the scope lines to the left of the gap but cannot be inside any extra ones; that is, all lines to the left of the resource must continue to the left of the gap.

The right side of the figure illustrates the results of exploiting the conjunction. When we exploit it, we add its components as new resources at the top of the gap. If either component of the conjunction is already among the active resources of the gap, this component *need* not be added again, but there is nothing wrong with doing so. The number *n* of this stage in the development of the derivation is written to the right of the conjunction to show that it has been exploited at this stage, and the stage number is also shown, along with the label Ext, to the left of the two lines that are added. Once the conjunction has been exploited, it is no longer an active resource for this gap though it could be active in other gaps (we will see later how to tell). The numbers in a derivation thus record the order of its development and also provide a way of telling when and where resources are exploited.

These numbers are also one of the devices derivations use to encode the structure of tree-form proofs: they mark the relation between premises and conclusion that tree-form proofs mark by a horizontal line. In English argumentation, words and phrases like therefore, hence, and it follows that indicate the same sorts of connections though in a less explicit way.

Another way to narrow a gap is to restate the problem it represents so that the goal we seek to reach is replaced by one or more simpler goals. We will call this process ***goal planning***. The law for conjunction as a conclusion tells us how we may plan for a goal that is a conjunction. Such a goal is entailed by our active resources if and only if each of its components is entailed. So the project of reaching a conjunction φ ∧ ψ from given resources comes to the same thing as completing two projects—namely, reaching each of the components φ and ψ from those same resources. This sort of goal planning thus uses Conjunction and takes the form shown in Figure 2.2.3-2.

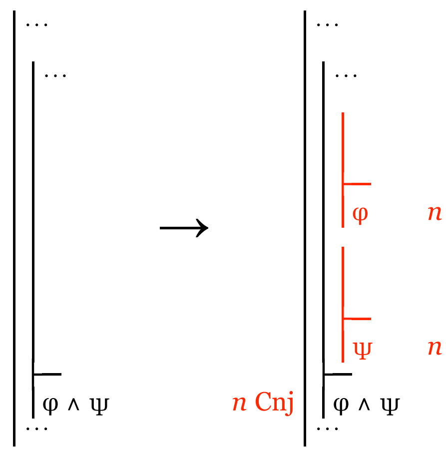



Fig. 2.2.3-2. Developing a derivation by planning for a conjunction at stage *n*.

On the left, no assumptions are made about the resources, but the goal is shown as a conjunction. On the right, we have introduced two new gaps, each with one of the conjunction's components as its goal. The two new goals bring with them two scope lines and are marked off by horizontal lines (as was the initial conclusion) to show that they represent the new material that led to the use of new scope lines. At the right of each of the new goals is a number showing the stage at which it was added. The same number appears to the left of the goal along with the label Cnj.

While in the case of Ext, numbers appeared at the left of the resources that were added and at the right of the resource being exploited, numbers here appear on the right of the new goals and at the left of the old one. This is because the new goals are introduced as premises from which the old one may be concluded while the resources added by Ext are added as conclusions drawn from the resource that is exploited; but, in both cases, the numbers mark a connection between premises and conclusions. The numbers here also serve, as do those for Ext, to show that an element of the derivation has been superceded by new additions. But, in the case of Cnj, this information is also provided in other ways: a gap will always have exactly one goal, and that goal will appear immediately below it.

The new gaps introduced in planning for a conjunction initially have the same active resources as the original gap. As resources are exploited in narrowing one of the gaps, these resources will become inactive for that gap; but they will remain active for the other gap until they are exploited there. When a derivation contains more than one gap, the question of where resources are active becomes important, and something will be said about it shortly. But, when we are dealing with conjunction alone, it is possible to mimic the procedure used for tree-form proofs and exploit the initial resources completely before we plan for goals. As a result, a general discussion of active and inactive resources can be postponed until we have considered an actual example of a derivation.

What we cannot postpone is an account of how a gap may be closed. If the goal of a gap appears also among its resources, the law for premises tells us that the goal is entailed by these resources, and the gap may be closed. The rule we use to do this is shown in Figure 2.2.3-3 below.

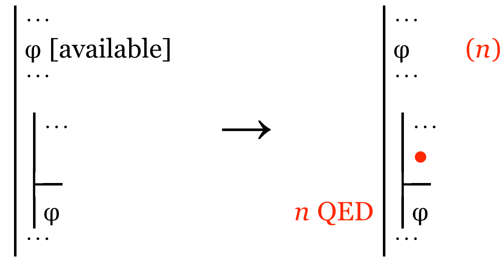


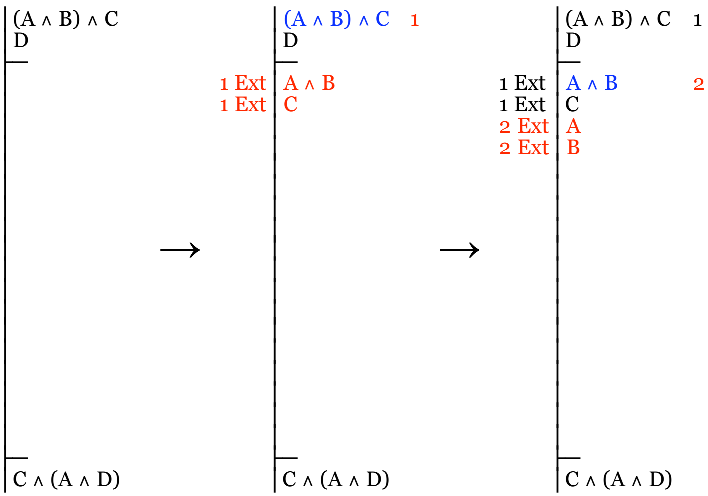

Fig. 2.2.3-3. Closing a gap by locating its goal among its resources.

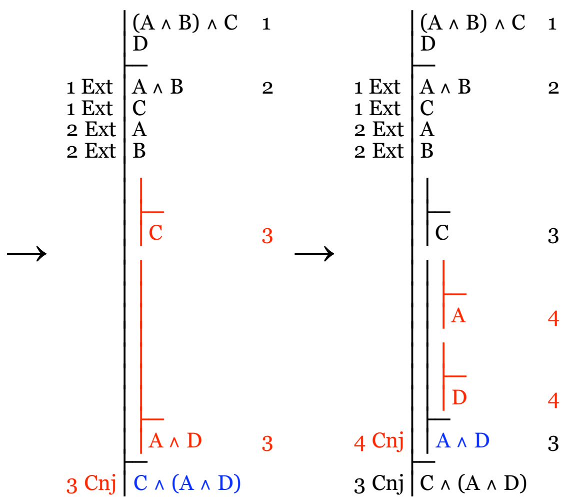
The label for this rule abbreviates the Latin phrase quod erat demonstrandum, which might be translated as what was to be proven. This Latin phrase is traditionally used when a planned conclusion is reached.

The stage number appears to the left of the goal (along with the label) since the goal is the conclusion, and it appears to the right of the resource since the resource is the premise. The latter number is enclosed in parentheses to indicate that the premise is not here being exploited. Since the gap is closed, the question whether a resource is active or not becomes moot, but this sort of notation will be used later in other cases where resources are used without being replaced by simpler resources of equivalent content; and QED shares with these rules the feature that the resources to which it is applied do not need to be active. To make it easy to see that the gap is now closed, it is filled with the symbol • (a ***black circle***). This is really not part of the derivation itself and is not given a stage number; it instead functions like stage numbers to indicate the organization of a derivation. Think of an analogy with written language: the symbol • marks the end of a series of stages in the way a period marks the end of a series of words.

### 2.2.4. An example

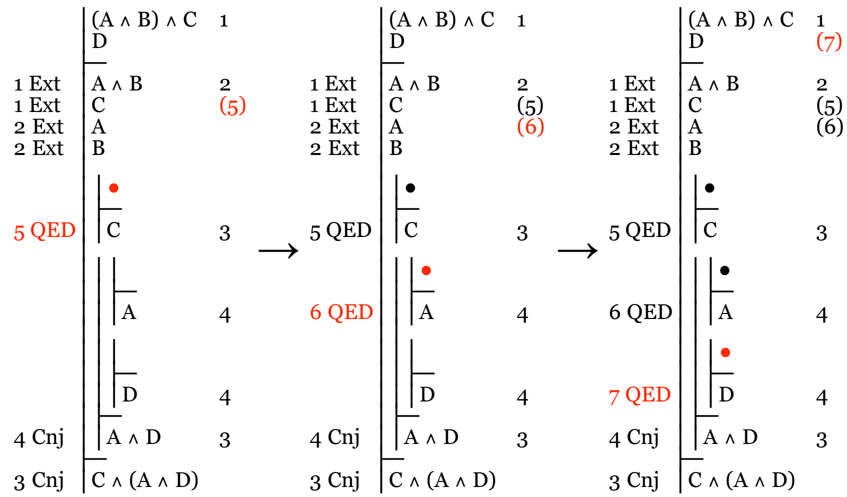
Now, let us restate an example for which we used tree-form proofs, now using the notation for derivations. The development is shown stage by stage below. At each stage, new material is shown in red. Resources that are exploited or goals that are planned for are shown in blue. At each of the stages 1 and 2, a resource is exploited. The added resources are conclusions drawn from the exploited resource, so the number of the stage is written at the left of the resources that are added and at the right of the one that is exploited.





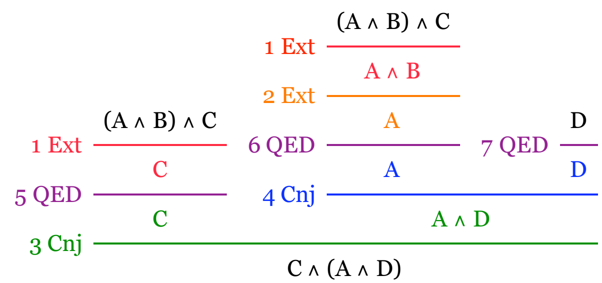
In stages 3 and 4, we plan for goals. The goals we add in each case are premises from which we plan to conclude the one they replace. The stage number therefore appears at the right of the new goals and to the left of the one we plan for.

In the last three stages we close gaps.



No sentences are added and the stage numbers merely mark the connection between resources that serve as premises and the goals that are concluded from them.

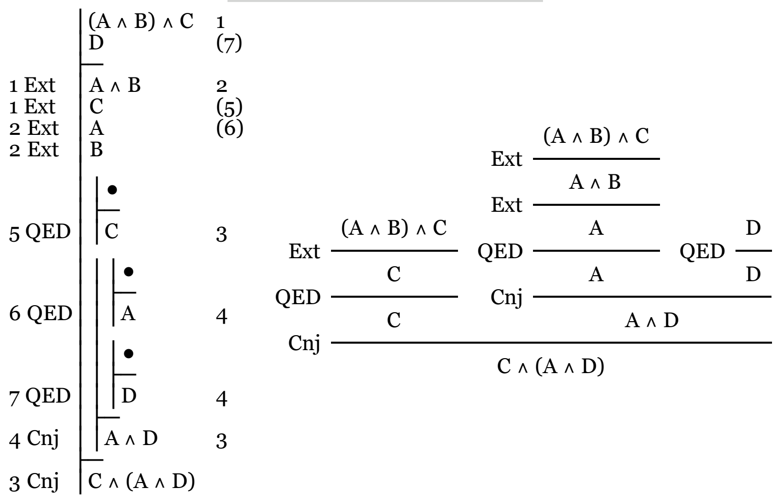
The proof tree from the original example is shown below with corresponding stage numbers added and with colors used to group items added at the same stage.



In stages 1 and 2, the new sentences lie below the horizontal line that is added because they are conclusions we draw as we move down chains of conclusions. In stages 3 and 4, the new sentences are above the line because they are premises from which we plan to reach the conclusion lying below them in a tree-form proof. And in the last three stages only the line is added because we are merely connecting conclusions we have accumulated to premises we have found we need. We use the label QED here, treating it as a pattern of argument whose conclusion is its only premise.

The diagram below shows the derivation and tree-form proof side by side. If your browser has JavaScript enabled, it can be used to display the state of both the derivation and the tree-form proof stage by stage. Simply place the cursor over the number of each stage in turn. (0 is used as the number of the initial stage.) The elements of the tree-form proof are shown in their final location, so the premises appear (with the first one repeated) above the points where they are eventually connected to the branches growing up from the conclusion.

0 1 2 3 4 5 6 7




One difference between the two proofs appears at stage 2, when the resources A and B are added to the derivation while only A appears as a conclusion in the tree-form proof. This is because a derivation leads us to accumulate as many conclusions as possible from our premises and B is one that is never needed to reach the final conclusion, something that is shown by the fact that no number appears to its right.



### 2.2.5. More rules

The principles Ex Nihilo Verum and Ex Falso Quodlibet appear in derivations as rules for closing gaps. In the case of the first, for a gap to be closed it is enough that it have ⊤ as its goal. No resource is involved, and the stage number appears only as an annotation to the goal.

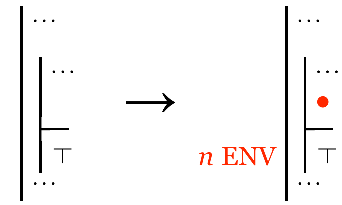



Fig. 2.2.5-1. Closing a gap that has ⊤ as its goal.

The rule EFQ takes a form much like QED.

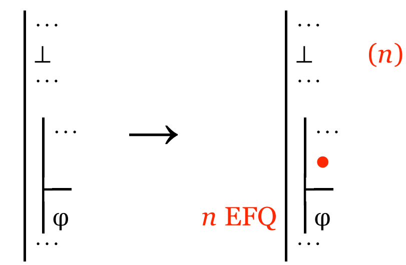
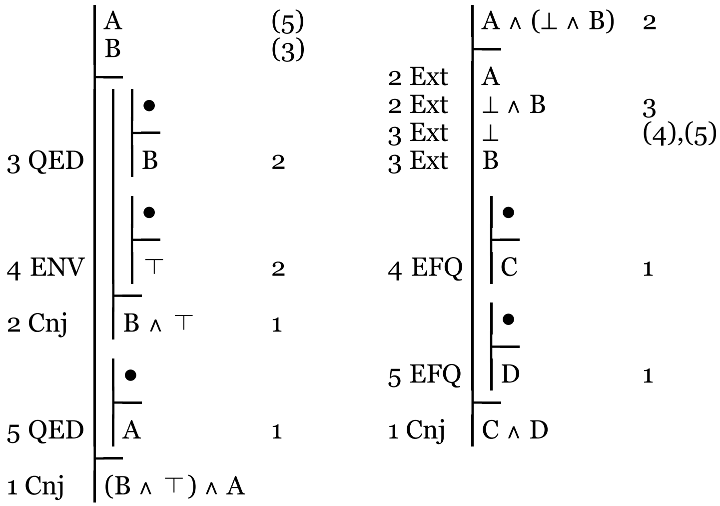



Fig. 2.2.5-2. Closing a gap that has ⊥ among its resources.

The difference is that having ⊥ as a resource enables us to close a gap no matter what its goal is. (If the goal also was ⊥, either EFQ or QED could be used.)

In 2.2.1, ENV and EFQ were illustrated by using tree-form proofs to show that A, B ⇒ (B ∧ ⊤) ∧ A and that A ∧ (⊥ ∧ B) ⇒ C ∧ D. As derivations, these proofs take the following forms:




Notice that, while every stage number of the second derivation appears somewhere among the annotations on its right-hand side, the same is not true of the first derivation because stage 4 is missing. Of course, that's because stage 4 is when we used ENV, and ENV is a valid argument without premises. Since stage numbers appearing in annotations on the left-hand side of a derivation correspond to horizontal lines lying above conclusions in the tree-form proof and ENV is the only form of argument so far without premises, it should be the only reason for a stage number to appear on the left but not the right.

You can use this idea as a way of checking for errors, and there are some further generalizations like this that you can use as checks. We will have no rules without conclusions, so every stage number should appear somewhere in the left-hand annotations. And, in a completed derivation whose gaps all close, all sentences other than assumptions (which, for now, are just the initial premises) will be conclusions and thus should have annotations on their left-hand side. Resources that are never used may appear with no annotations on their right; but, as you are constructing a derivation, it can be very useful to check for the absence of right-hand annotations because this can lead you to notice resources that you have not yet exploited. And, when we go on (in 2.3) to use derivations to show that claims of entailment fail, a check for the absence of right-hand annotations will be the key test of whether we done everything possible to complete a derivation.



### 2.2.6. Resources

The ideas of available and active resources have been used at several points already, but they have not yet been explained fully. A resource counts as ***available*** in a gap if it was entered either as one of the initial premises of the derivation or in the course of developing the gap in question. The system of scope lines can be used to tell which resources are available in a gap: a resource is available if every scope line to its left continues unbroken at the left of the gap.

One way of thinking about this is shown in Figure 2.2.6-1.

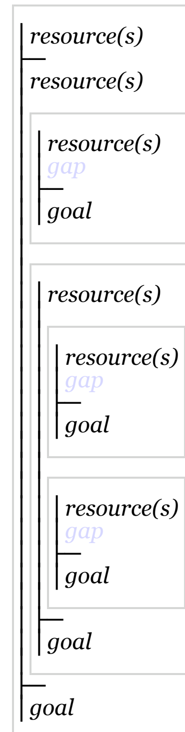
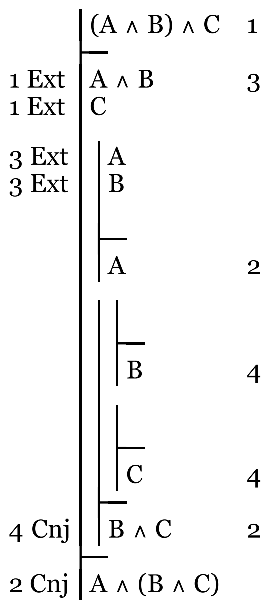



Fig. 2.2.6-1. The boxes indicated by the scope lines of a derivation. If JavaScript is enabled on the browser you are using, moving the cursor over a resource will color the gaps in which it is available green and shade areas where it is unavailable. Moving the cursor over a gap will color resources available in it green and shade areas whose resources are unavailable to it. The resource or gap that the cursor is over will be colored blue and underlined.

You may suppose that each scope line indicates the left side of a box and that a resource is available only to the gaps that are also within the smallest box containing it.

A resource is ***active*** in a gap if it is available in that gap and has not already been exploited in narrowing it. The easiest way to locate the active resources of a gap is to scan the available resources and eliminate the inactive ones. To be inactive in any gap, a resource must have been exploited at some stage. If it has, there will be an unparenthesized stage number to its right. A resource may have been exploited only in some gaps and may still remain active in others. To be inactive in a given gap, the resource must have been exploited in narrowing the gap. To see whether this is so, we need to check all resources and goals that were introduced at a stage when the resource was exploited (i.e., at a stage whose number appear unparenthesized to the resource's right). (So far, we have seen goals introduced only in the course of planning for more distant goals, but in later chapters they will be introduced as part of the exploitation of certain resources.) If any such resource or goal is such that the smallest box containing it also contains the gap we are considering, it was introduced in the course of developing the gap. A resource may be exploited more than once, so there may be several stage numbers you will need to check. If any of them was a stage in which the gap you are considering was developed, the resource is no longer among the active resources of the gap.

This is illustrated by the partially developed derivation shown below.




The three steps at the top of the derivation are resources available for each of the derivation's three gaps. The first, (A ∧ B) ∧ C, is inactive in all three gaps. It was exploited at stage 1, and that was the initial stage of development for all the gaps of the derivation. The second resource, A ∧ B, is inactive for the first of the gaps (having been exploited at stage 3 in developing this gap), but it is active for the remaining two gaps since the resources introduced at stage 3 did nothing to narrow these gaps (as is shown by the fact that the gaps are outside the smallest box surrounding the resources with 3 at their left). The third resource C has not been exploited at all (and could not be since it is not a conjunction), so it is active for all three gaps. Since the resource exploited at stage 3 must be exploited again in order to close the second gap, it would have been a little more efficient to exploit this resource before dividing the initial gap in two; but the derivation as shown is perfectly correct (though still unfinished).

You may suppose that a given gap can see only those parts of a derivation that are not boxed off from it—i.e., only those parts all of whose scope lines continue to the left of the gap. If a stage number appears at the left only in parts of the derivation that are invisible to the gap, it is also invisible—even when it appears to the right of resources that are visible.

This idea is illustrated in Figure 2.2.6-2 below where the same derivation is shown from the perspective of each of the three gaps in turn.

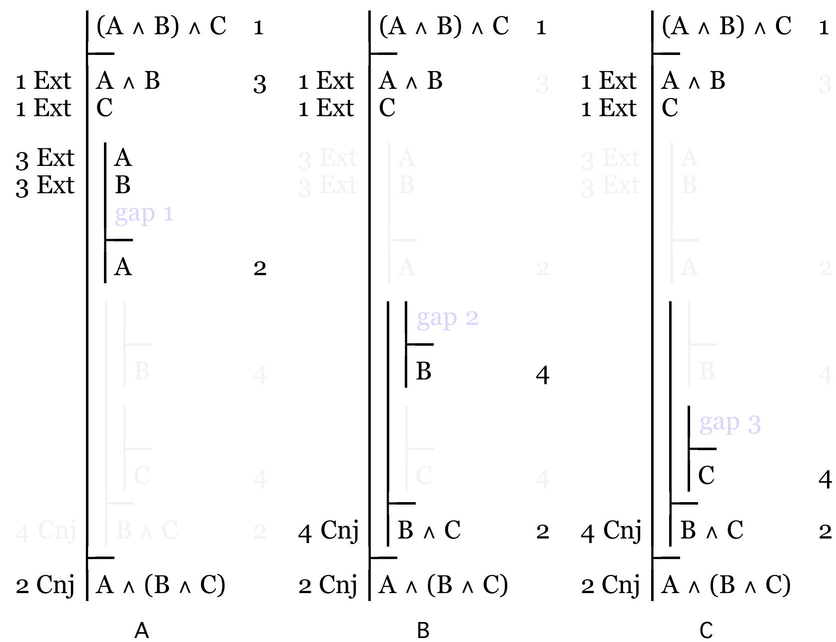



Fig. 2.2.6-2. A derivation from the perspective of each of its three gaps.

Material that is boxed off from a gap is shown in light gray. Notice that the number 3 at the right of the second line is invisible to the second and third gaps. As we saw earlier, that is because all the development at stage 3 is boxed off from the second and third gaps.

Any derivation can be thought of as the result of superimposing separate layers like these. There will be one layer for each gap with a gap's layer depicting its perspective on the derivation. When we distinguish the resources available for a gap or determine whether a resource has been used to narrow a gap, we are really considering that gap's layer separately.

When a gap is divided before a resource is exploited to narrow it, it is possible to exploit the resource to narrow several gaps at once. This is shown in the partial derivation below (which has the same initial premises and conclusion as the one we have been considering).

| | | |
|---|---|---|
| | (A ∧ B) ∧ C | 1 |
| 1 Ext | A ∧ B | 4 |
| 1 Ext | C | |
| 4 Ext | A | |
| 4 Ext | B | |
| | A | 2 |
| 4 Ext | A | |
| 4 Ext | B | |
| | B | 3 |
| | C | 3 |
| 3 Cnj | B ∧ C | 2 |
| 2 Cnj | A ∧ (B ∧ C) | |

In this derivation, one of the resources has just been exploited at stage 4 to narrow two different gaps. Thereafter, it is inactive in these gaps but still active in the third (where it happens to be unneeded). Some of the resources added at stage 4 will be invisible to each of the first two gaps; but, because other added resources are visible, the number 4 at the right is visible from both these gaps. However, none of the resources added at stage 4 is visible from the third gap, so the number 4 at the right is not visible from it.

Since we use a similar numerical notation for both resources that are exploited and goals that have been planned for, you might expect that the concepts of availability and activity can be applied to goals as well as resources; and, indeed, they can be. If we were to consider derivations for relative exhaustiveness, we would need to engage in the same sort of accounting for goals that we have been considering for resources. However, in a system of derivations for entailment alone like the one we will actually use, each gap has just one active goal, which appears just below the gap. Goals at earlier stages of a gap's development (i.e., the goals that are not boxed off from the gap) could be described as "available", but they are not available for any sort of use. In particular, although we can consider all available resources when looking for a way of closing a gap, it is only the active goal and not any earlier one that we consider. (Some of the arguments of 2.3.2 could be used to show that considering all "available" goals would not lead us to count an invalid argument as valid, but looking at derivations in this way would make them less like the patterns of ordinary explicit deductive argumentation, which seem to be focused always on a single conclusion.)

## 2.2.s. Summary

1 One way of combining principles of entailment uses the graphical idea of a tree. This provides a natural notation for the patterns of argument conjunction and extraction. The adequacy of this approach to entailment concerning conjunction can be shown by considering principles of entailment that state conditions for the validity of arguments that have conjunctions as conclusions or as premises. And patterns of argument, also with Latin names, can be added to capture the properties of ⊤ (ex nihilo verum) and ⊥ (ex falso quodlibet).

2 In fact, we will use a different, more compact notation for combining principles of entailment—a kind of natural deduction system that we will refer as a system of derivations. This notation presents the project of showing that an entailment holds as the task of closing a gap between its conclusion, which serves as a goal, and its premises, which serve as resources. As we narrow the initial gap (and others that result from it), we develop the derivation. The branching structure of tree-form proofs is represented in part by a system of vertical scope lines and in part by numerical annotations.

3 The laws of entailment appear as rules for exploiting resources, planning for goals, and closing gaps. There are rules for each of the patterns of argument that figure in tree-form proofs. The key rules for conjunction are Extraction (Ext) and Conjunction (Cnj). Quod Erat Demonstrandum (QED) is used to close a gap when its goal is among its resources, and the symbol • (a black circle) marks a closed gap.

4 When a derivation is developed, numbers are used along with the labels for rules to record both the order of the development and the connection between the premises and conclusions of the rules.

5 Principles of entailment for other logical forms will be associated with further rules. Those for ⊤ and ⊥ are the rules Ex Nihilo Verum (ENV) and Ex Falso Quodlibet (EFQ).

6 We keep track of changes in the information contained in goals and resources by using the scope lines of a derivation to tell in which gaps given resources are available and in which gaps available resources are still active.



## 2.2.x. Exercise questions

**1.** Restate the derivation below as a tree-form proof, labeling each horizontal line with the number of the stage at which it is entered. That is, do what is done with the example in 2.2.4

| | | |
|---|---|---|
| | (A ∧ C) ∧ B | 1 |
| 1 Ext | A ∧ C | 2 |
| 1 Ext | B | (4) |
| 2 Ext | A | |
| 2 Ext | C | (5) |
| | \| • | |
| 4 QED | \| B | 3 |
| | \| • | |
| 5 QED | \| C | 3 |
| 3 Cnj | B ∧ C | |

**2.** Use the system of derivations to establish each of the following claims of entailment:

**a.** A ∧ B ⇒ B ∧ A

**b.** A ⇒ A ∧ A

**c.** A ∧ (B ∧ C) ⇒ (C ∧ B) ∧ A

**d.** A, B ∧ C, D ⇒ (C ∧ (B ∧ A)) ∧ B
[The derivation for **d** will have three premises above the initial horizontal line.]

**e.** A ∧ (B ∧ C) ⇒ (B ∧ A) ∧ (C ∧ A)

## 2.2.xa. Exercise answers

**1.**

$$\dfrac{\text{4 QED}\ \dfrac{\text{1 Ext}\ \dfrac{(A \wedge C) \wedge B}{B}}{B} \qquad \text{5 QED}\ \dfrac{\text{2 Ext}\ \dfrac{\text{1 Ext}\ \dfrac{(A \wedge C) \wedge B}{A \wedge C}}{C}}{C}}{B \wedge C}\ \text{3 Cnj}$$

**2.** **a.**

| | | |
|---|---|---|
| | │ A ∧ B | 1 |
| | ├─ | |
| 1 Ext | │ A | (4) |
| 1 Ext | │ B | (3) |
| | │ │ • | |
| | │ ├─ | |
| 3 QED | │ │ B | 2 |
| | │ │ • | |
| | │ ├─ | |
| 4 QED | │ │ A | 2 |
| | ├─ | |
| 2 Cnj | │ B ∧ A | |

**b.**

| | | |
|---|---|---|
| | │ A | (2),(3) |
| | ├─ | |
| | │ │ • | |
| | │ ├─ | |
| 2 QED | │ │ A | 1 |
| | │ │ • | |
| | │ ├─ | |
| 3 QED | │ │ A | 1 |
| | ├─ | |
| 1 Cnj | │ A ∧ A | |

**c.**

| | | |
|---|---|---|
| | │ A ∧ (B ∧ C) | 1 |
| | ├─ | |
| 1 Ext | │ A | (7) |
| 1 Ext | │ B ∧ C | 2 |
| 2 Ext | │ B | (6) |
| 2 Ext | │ C | (5) |
| | │ │ │ • | |
| | │ │ ├─ | |
| 5 QED | │ │ │ C | 4 |
| | │ │ │ • | |
| | │ │ ├─ | |
| 6 QED | │ │ │ B | 4 |
| | │ ├─ | |
| 4 Cnj | │ │ C ∧ B | 3 |
| | │ │ • | |
| | │ ├─ | |
| 7 QED | │ │ A | 3 |
| | ├─ | |
| 3 Cnj | │ (C ∧ B) ∧ A | |

**d.**

| | | |
|---|---|---|
| | │ A | (7) |
| | │ B ∧ C | 1 |
| | │ D | |
| | ├─ | |
| 1 | │ B | (6) |
| 1 | │ C | (5) |
| | │ │ │ • | |
| | │ │ ├─ | |
| 5 QED | │ │ │ C | 3 |
| | │ │ │ │ • | |
| | │ │ │ ├─ | |
| 6 QED | │ │ │ │ B | 4 |
| | │ │ │ │ • | |
| | │ │ │ ├─ | |
| 7 QED | │ │ │ │ A | 4 |
| | │ │ ├─ | |
| 4 Cnj | │ │ │ B ∧ A | 3 |
| | │ ├─ | |
| 3 Cnj | │ │ C ∧ (B ∧ A) | 2 |
| | │ │ • | |
| | │ ├─ | |
| | │ │ B | 2 |
| | ├─ | |
| 2 Cnj | │ (C ∧ (B ∧ A)) ∧ B | 2 |

**e.**

| | | |
|---|---|---|
| | $A \wedge (B \wedge C)$ | 1 |
| 1 Ext | $A$ | (7),(9) |
| 1 Ext | $B \wedge C$ | 2 |
| 2 Ext | $B$ | (6) |
| 2 Ext | $C$ | (8) |
| | • | |
| 6 QED | $B$ | 4 |
| | • | |
| 7 QED | $A$ | 4 |
| 4 Cnj | $B \wedge A$ | 3 |
| | • | |
| 8 QED | $C$ | 5 |
| | • | |
| 9 QED | $A$ | 5 |
| 5 Cnj | $C \wedge A$ | 3 |
| 3 Cnj | $(B \wedge A) \wedge (C \wedge A)$ | |