7

Natural Deduction for Sentence Logic

Derived Rules and Derivations without Premises

7–1. DERIVED RULES

This section begins with a somewhat strange example. We will first follow our noses in putting together a derivation using the strategies I have recommended. When we are done, we will notice that some of the steps, although perfectly correct, do no real work for us. We will then find interesting ways to avoid the superfluous steps and to formulate in a general way some methods for making derivations easier.

Let's derive 'A⊃(B⊃C)' from '(A⊃B)⊃C'. Our derivation must begin like this:

```
1 | (A⊃B)⊃C    P
  |--------
  | ?
  |
  | ?
  |
  | A⊃(B⊃C)
```


We will pursue the obvious strategy of getting the conclusion by constructing a subderivation from the assumption of 'A' to 'B⊃C' as conclusion:

```
1 | (A⊃B)⊃C      P
  |--------
2 |   | A        A
  |   |---
  |   | ?
  |   |
  |   | ?
  |   | B⊃C
  | A⊃(B⊃C)      ⊃I
```

We have reduced our task to that of deriving 'B⊃C' from 'A', where we can use the outer derivation's premise. But how are we going to do that?

The target conclusion we now need to derive, 'B⊃C', is itself a conditional. So let's try setting up a sub-sub-derivation with 'B' as assumption from which we are going to try to derive 'C'. We are shooting for a derivation which looks like this:

```
1 | (A⊃B)⊃C        P
  |--------
2 |   | A          A
  |   |---
3 |   |   | B      A
  |   |   |---
  |   |   | ?
  |   |   | ?
  |   |   | C
  |   | B⊃C        ⊃I
  | A⊃(B⊃C)        ⊃I
```

How are we going to squeeze 'C' out of 'B'? We have not yet used our premise, and we notice that the consequence of the premise is just the needed sentence 'C'. If only we could also get the antecedent of the premise, 'A⊃B', in the sub-sub-derivation, we could use that and the premise to get 'C' by ⊃E.

It might look rough for getting 'A⊃B' into the sub-sub-derivation, but once you see how to do it, it's not hard. What we want is 'A⊃B', which, again, is a conditional. So we will have to start a sub-sub-sub-derivation with 'A' as assumption where we will try to get 'B' as a conclusion. But that's easy because this sub-sub-sub-derivation is a subderivation of the derivation with 'B' as its assumption. So all we have to do is reiterate 'B' in our sub-sub-sub-derivation.

If this is a little confusing, follow it in the completed derivation below, rereading the text if necessary to see clearly the thought process which leads me along step by step:

```
 1 | (A⊃B)⊃C            P
   |---------
 2 |   | A              A
   |   |----
 3 |   |   | B          A
   |   |   |----
 4 |   |   | (A⊃B)⊃C    1, R
 5 |   |   |   | A      A
   |   |   |   |----
 6 |   |   |   | B      3, R
   |   |   | 4
 7 |   |   | A⊃B        5-6, ⊃I
 8 |   |   | C          4, 7, ⊃E
   |   | 3
 9 |   | B⊃C            3-8, ⊃I
   | 2
10 | A⊃(B⊃C)            2-9, ⊃I
```

I've carefully gone through this example for you because I wanted to illustrate again our strategies for constructing derivations. In this case, though, we have produced a derivation which, while perfectly correct, has an odd feature. Notice that I got 'B' in step 6 by just reiterating it. I never used the assumption, 'A'! In just the same way, I never used the assumption of 'A' on line 2 in deriving 'B⊃C' in line 9. The fact that 'A' was assumed (twice no less!), but never used, in no way shows anything technically wrong with the derivation. Any derivation correctly formed, as this one is, following the rules, counts as correct even if it turns out that parts were not used. No one ever said that a line, either an assumption or a conclusion, **has** to be used.

I should refine what I just said: The assumptions of 'A', both in line 2 and in line 5, were not used in deriving the target conclusions of the subderivations in which 'A' was assumed. But we had to assume 'A' in both cases to permit us to apply ⊃I to introduce a conditional with 'A' as the antecedent. However, if in subderivation 2 the assumption 'A' was never used in deriving 'B⊃C', you would suspect that we could derive not just 'A⊃(B⊃C)' but the stronger conclusion 'B⊃C' from the original premise. And, indeed, we can do just that:


```
1 | (A⊃B)⊃C           P
  |---------
2 |   | B             A
  |   |----
3 |   | (A⊃B)⊃C       1, R
4 |   |   | A         A
  |   |   |----
5 |   |   | B         2, R
6 |   | A⊃B           4, 5, ⊃I
7 |   | C             3, 6, ⊃E
8 | B⊃C               2-7, ⊃I
```

Now we notice that we could have worked the original problem in a different way. We could have first derived 'B⊃C', as I have just done. Then we could have modified this derivaton by inserting the subderivation beginning with 'A', the subderivation 2 in the previous derivation, and then applying ⊃I. In other words, if we can derive 'B⊃C', we can always derive 'A⊃(B⊃C)' by simply assuming 'A', deriving 'B⊃C' by whatever means we used before, and then applying ⊃I. In fact, we can organize things most clearly by separating the two steps. First derive 'B⊃C', then create a subderivation with assumption 'A' and conclusion 'B⊃C' obtained by reiterating 'B⊃C' from above. Then apply ⊃I. The relevant parts of the total derivation, beginning with the previously derived conclusion, 'B⊃C', will look like this:

```
| .
| .
| .
| B⊃C
|   | A        A
|   |----
|   | B⊃C      R
| A⊃(B⊃C)      ⊃I
```

We have just discovered something extremely interesting: Nothing in the above line of thought turns on the two sentences involved being 'B⊃C' and 'A'. This procedure will work for any old sentences **X** and **Y**. For any sentences **X** and **Y**, if we can derive **Y**, we can always extend the derivation to a new derivation with conclusion **X⊃Y**. If

```
| .
| .
| .
| Y
```

stands for the part of the derivation in which we derive **Y**, the new derivation will look like this:

```
|  .
|  .
|  .
|  Y
|     | X    A
|     |---
|     | Y    R
|
|  X⊃Y       ⊃I
```

Because **X** and **Y** can be any sentences at all, we can express the fact we have just discovered as a "new" rule of inference:

```
| [Y]
|
| (X⊃Y)  Weakening (W)
```

In other words, if any sentence, **Y**, appears in a derivation, you are licensed to write **X⊃Y** anywhere below, using any sentence you like for **X**. This rule of inference is not really new (that's why a moment ago I put quotes around the word "new"). If we want to, we can always dispense with the weakening rule and use our original rules instead. Wherever we have used the weakening rule, we can always fill in the steps as shown above by assuming **X**, reiterating **Y**, and applying ⊃I.

Dispensable, shortcut rules like weakening will prove to be extraordinarily useful. We call them *Derived Rules.*

> A *Derived Rule* is a rule of inference which can always be replaced by some combination of applications of the original rules of inference. The original rules are called the *Primitive Rules* of inference.
>
> A proof of a derived rule is a demonstration which shows how the derived rule may be systematically replaced by application of the primitive rules of inference.

The weakening rule is proved in the schematic derivation which you saw immediately above.

By using the derived weakening rule, we can immensely simplify the derivation we have been studying in this section. For we can use weakening instead of both of the subderivations which start from 'A' as an assumption. In addition to the use of weakening which we have already seen, we can use it in the subderivation which begins by assuming 'B'. Given 'B' as an assumption, weakening immediately licenses us to write 'A⊃B', which we need for applying ⊃E.

```
1 | (A⊃B)⊃C          P
  |---
2 |    | B           A
  |    |---
3 |    | (A⊃B)⊃C     1, R
4 |    | A⊃B         2, W
5 |    | C           3, 4, ⊃E
  |
6 | B⊃C              2–5, ⊃I
7 | A⊃(B⊃C)          6, W
```

Isn't that easy!

7–2. ARGUMENT BY CASES

Once we see how much work derived rules can save us, we will want others. Indeed, many derivations which are prohibitively complicated if we use only primitive rules become quite manageable when we can use derived rules also. Here is another example of a derived rule:

Argument by Cases

```
| [XvY]
| [X⊃Z]
| [Y⊃Z]
|
| (Z)      AC
```

Here is a proof of this derived rule:

```
     .
     .
     .
1  | XvY        (Input for the derived rule)
2  | X⊃Z        (Input for the derived rule)
3  | Y⊃Z        (Input for the derived rule)
4  |   | ~Z     A
   |   |---
5  |   |   | X         A
   |   |   |---
6  |   |   | X⊃Z       2, R
7  |   |   | Z         5, 6, ⊃E
8  |   |   | ~Z        4, R
   |   3
9  |   | ~X            5–8, ~I
10 |   | XvY           1, R
11 |   | Y             9, 10, vE
12 |   | Y⊃Z           3, R
13 |   | Z             11, 12, ⊃E
   |   2
14 | ~~Z               4–13, ~I
15 | Z                 14, ~E
   1
```

Again, the point of this proof is this: Suppose you have just used the derived rule *Argument by Cases*. Then, if you really want to, you can go back and rewrite your derivation using only the primitive rules. This proof shows you how to do it. Whatever sentence you have used for **X** and whatever sentence you have used for **Y**, just substitute the above in your derivation, after having written in your sentences for **X** and **Y**. Of course, you will have to redo the line numberings to fit with those in your full derivation. I have put in line numbers above to help in talking about the proof.

(A small point to notice in this example: In line 14 I have appealed to subderivation 2, lines 4–13, to use negation introduction. But where in the subderivation did I conclude both a sentence and its negation? The point is that the assumption can count as one of these sentences. Why? Because any sentence can be derived from itself, by reiteration. Thus, in derivation 2, I can fairly enough count both **~Z** and **Z** as following from **~Z.**)

Here is another derived rule, which we will also call *Argument by Cases* because it differs only superficially from the last:

Argument by Cases (second form)

```
| XvY
| .
| .
| .
|    | X   A
|    |----
|    | .
|    | .
|    | .
|    | Z
| .
| .
| .
|    | Y   A
|    |----
|    | .
|    | .
|    | .
|    | Z
| Z       AC
```

In words, if in a derivation you already have a sentence of the form **XvY**, a subderivation from **X** as assumption to **Z** as conclusion, and a second subderivation from **Y** as assumption to **Z** as conclusion, you are licensed to write **Z** as a conclusion anywhere below.

This second form of argument by cases is actually the form you will use most often.

The proof of the second form of argument by cases goes like this:

```
1 | XvY         (Input for the derived rule)
  | .
  | .
  |    | X    A
  |    |----
  |    | .      (Input for the derived rule)
  |    | .
  |    | Z
2 | X⊃Z           ⊃I
  | .
  |    | Y    A
  |    |----
  |    | .      (Input for the derived rule)
  |    | .
  |    | Z
3 | Y⊃Z           ⊃I
  | Z           From lines 1, 2, 3 and the first form of
                Argument by Cases
```

Note that in this proof I have used a previously proved derived rule (the first form of argument by cases) in proving a new derived rule (the second form of argument by cases). Why should I be allowed to do that, given that a proof of a derived rule must show that one can dispense with the derived rule and use primitive rules instead? Can you see the answer to this question?

Suppose we want to rewrite a derivation which uses a new derived rule so that, after rewriting, no derived rules are used. First rewrite the derivation dispensing with the new derived rule, following the proof of the new derived rule. The resulting derivation may use previously proved derived rules. But now we can rewrite some more, using the proofs of the previously proved derived rules to get rid of them. We continue in this way until we arrive at a (probably horrendously long) derivation which uses only primitive rules.

Argument by cases is an especially important derived rule, so much so that in a moment I'm going to give you a batch of exercises designed exclusively to develop your skill in applying it. Its importance stems not only from the fact that it can greatly shorten your work. It also represents a common and important form of reasoning which gets used both in everyday life and in mathematics. In fact, many texts use argument by cases as a primitive rule, where I use what I have called disjunction elimination. In fact, given the other rules, what I have called argument by cases and disjunction elimination are equivalent. I prefer to start with my form of disjunction elimination because I think that students learn it more easily than argument by cases. But now that your skills have developed, you should learn to use both rules effectively.

EXERCISES

7–1. Use argument by cases as well as any of the primitive rules to construct derivations to establish the validity of the following arguments:

a) AvB / BvA

b) Av(BvC) / (AvB)vC

c) (AvB)&(B⊃C) / AvC

d) (A&B)v(A&C) / A&(BvC)

e) A&(BvC) / (A&B)v(A&C)

f) Av(B&C) / (AvB)&(AvC)

g) (AvB)&(AvC) / Av(B&C)

h) KvL, K≡L / K&L

i) (D⊃G)v(D⊃I) / D⊃(GvI)

j) ~CvK, A⊃D / (AvC)⊃(KvD)

k) ~HvM, ~M⊃~C / (HvC)⊃M

l) (S&J)v(~S&~J) / S≡J

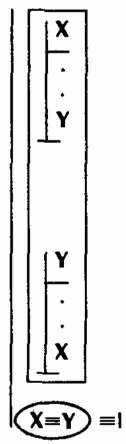
m) K⊃(FvC), J⊃(CvD), ~C / ~(FvD)⊃~(KvJ)

7–2. Show that in the presence of the other primitive rules, vE is equivalent to AC. (Half of this problem is already done in the text. Figure out which half and then do the other half!)

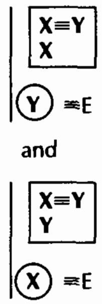
7–3. FURTHER DERIVED RULES

Here are some further derived rules. In naming these rules I have used the same name for similar or closely connected primitive rules.

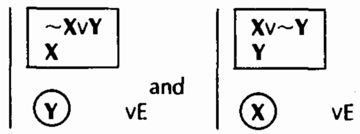
Biconditional Introduction



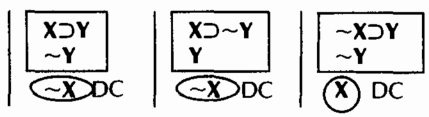
Biconditional Elimination



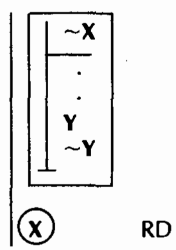
Disjunction Elimination



Denying the Consequent
(Traditionally called "Modus Tolens")



Reductio Ad Absurdum



The reductio ad absurdum rule looks like a negation elimination rule. Actually, as you will see when you do the exercises, it uses both ~I and ~E.

We can get further derived rules from the laws of logical equivalence (chapters 3 and 4). For example, any sentence of the form ~(XvY) is logically equivalent to **~X&~Y**. Because two logically equivalent sentences have the same truth value in any assignment of truth values to sentence letters, if one of these sentences validly follows from some premises and assumptions, so does the other. We can use these facts to augment our stock of derived rules:

$\boxed{\text{~(XvY)}}$		$\boxed{\text{~X\&~Y}}$	
~X&~Y	DM	~(XvY)	DM

Similarly, one can use other logical equivalences to provide derived rules. Here is a list of such rules you may use, given in a shortened notation. You should understand the first line as the pair of derived de Morgan rules immediately above. Understand the following lines similarly.

DE MORGAN'S RULES

~(**XvY**) and ~**X**&~**Y** are mutually derivable (DM).
~(**X&Y**) and ~**X**v~**Y** are mutually derivable (DM).

CONTRAPOSITION

X⊃Y and ~**Y**⊃~**X** are mutually derivable (CP).
~**X⊃Y** and ~**Y⊃X** are mutually derivable (CP).
X⊃~**Y** and **Y**⊃~**X** are mutually derivable (CP).

CONDITIONAL RULES

X⊃Y and ~**XvY** are mutually derivable (C).
~(**X⊃Y**) and **X**&~**Y** are mutually derivable (C).

The letters in parentheses are the abbreviations you use to annotate your use of these rules.

We could add further rules of mutual derivability based on the distributive, associative, commutative, and other laws of logical equivalence. But in practice the above rules are the ones which prove most useful.

It is not hard to prove that these rules of mutual derivability follow from the primitive rules—in fact, you will give these proofs in the exercises.

Many texts use rules of logical equivalence to augment the rules for derivations in a much more radical way. These strong rules of replacement, as they are called, allow you to substitute one logical equivalent for a subsentence **inside** a larger sentence. Thus, if you have derived '(AvB)⊃C', these strong rules allow you to write down as a further conclusion '(Av~~B)⊃C', where you have substituted '~~B' for the logically equivalent 'B'.

By the law of substitution of logical equivalents, we know that such rules of replacement must be correct, in the sense that they will always take us from true premises to true conclusions. But it is not so easy to prove these replacement rules as **derived** rules. That is, it is hard to prove that one can always systematically rewrite a derivation using one of these replacement rules as a longer derivation which uses only primitive rules. For this reason I won't be using these replacement rules. Your instructor may, however, explain the replacement rules in greater detail and allow you to use them in your derivations. Your instructor may also choose to augment the list of logical equivalents you may use in forming such rules.

EXERCISES

7–3. Prove all the derived rules given in the text but not proved there. In each of your proofs use only those derived rules already proved.

7–4. Provide derivations which establish the validity of the following arguments. You may use any of the derived rules. In fact, many of the problems are prohibitively complex if you don't use derived rules!

a) M&(~BvC)
——
B⊃C

b) M⊃(D⊃P)
M⊃D
——
M⊃P

c) (K⊃S)⊃(S⊃H)
S
——
K⊃H

d) F⊃O
L⊃J
——
(FvL)⊃(OvJ)

e) ~[(F&H)v~F]
——
~H

f) B⊃(HvR)
——
B⊃(~H⊃R)

g) ~(~M&D)
F⊃~M
Fv~D
——
~D

h) (A⊃B)&(D⊃~B)
(CvD)&(C⊃~B)
——
~A

i) P⊃(DvM)
——
(P⊃D)v(P⊃M)

j) (G&~M)⊃(~M&K)
K⊃~G
——
G⊃M

k) AvB
~B≡(CvD)
(D&E)v[D&(F⊃G)]
——
A

l) S≡J
——
(S&J)v(~S&~J)

m) ~C⊃[Fv~(DvN)]
~N⊃D
——
~F⊃C

n) (GvA)⊃(H⊃B)
[H⊃(H&B)]⊃K
——
G⊃K

o)
```
F⊃(KvB)
(~FvG)&(~Gv~K)
--------------
F⊃B
```

p)
```
Dv(M⊃J)
[M⊃(M&J)]⊃(PvK)
(P⊃D)&(K⊃F)
---------------
DvF
```

q)
```
Q≡~(A&F)
~(MvA)⊃~H
~(Q&A)vF
---------
Q⊃(H⊃M)
```

r)
```
(I&~T)⊃P
~A⊃~T
~TvC
C⊃D
-------------
~P⊃[I⊃(D&A)]
```

s)
```
B⊃(NvM)
N⊃(C&K)
C⊃(K⊃P)
~(P&B)
-------
B⊃M
```

7-4. DERIVATIONS WITHOUT PREMISES

When we discovered the derived weakening rule, we stumbled across the fact that a derivation (or a subderivation) does not have to use all, or even any, of its premises or assumptions. This fact is about to become important in another way. To fix ideas, let me illustrate with the simplest possible example:

```
0 | B           P
  |---
1 |    | A      A
  |    |---
2 |    | A      1, R
3 | A⊃A         1-2, ⊃I
```

The premise, 'B', never got used in this derivation. But then, as I put it before, who ever said that all, or even any, premises **have** to be used?

Once you see this last derivation, the following question might occur to you: If the premise, 'B', never gets used, do we have to have it? Could we just drop it and have a derivation with no premises? Indeed, who ever said that a derivation **has** to have any premises?

A derivation with no premises, which satisfies all the other rules I have given for forming derivations, will count as a perfectly good derivation. Stripped of its unused premise, the last derivation becomes:

```
1 |    | A      A
  |    |---
2 |    | A      1, R
3 | A⊃A         1-2, ⊃I
```


(You might now wonder: Can subderivations have no assumptions? We could say yes, except that an assumptionless subderivation would never do any work, for a subderivation helps us only when its assumption gets discharged. So I will insist that a subderivation always have exactly one assumption.)

All right—a derivation may have no premises. But what does a premiseless derivation mean?

Remember that the significance of a derivation with one or more premises lies in this: Any case, that is, any assignment of truth values to sentence letters, which makes all the premises true also makes all of the derivation's conclusions true. How can we construe this idea when there are no premises?

To approach this question, go back to the first derivation in this section, the one beginning with the premise 'B'. Since the premise never got used, we could cross it out and replace it by any other sentence we wanted. Let us indicate this fact symbolically by writing **X** for the premise, thereby indicating that we can write in any sentence we want where the '**X**' occurs

```
0 | X           P
  |---
1 |    | A      A
  |    |---
2 |    | A      1, R
3 | A⊃A         1-2, ⊃I
```

For example, for **X** we could put the logical truth, 'Av~A'. Because the result is a correct derivation, any assignment of truth values to sentence letters which makes the premise true must also make all conclusions true. But 'Av~A' is true for **all** cases. Thus the conclusion, 'A⊃A', must be true in all cases also. That is, 'A⊃A' is a logical truth. I can make the same point another way. I want to convince you that 'A⊃A' is true in all cases. So I'll let you pick any case you want. Given your case, I'll choose a sentence for **X** which is true in that case. Then the above derivation shows 'A⊃A' to be true in that case also.

Now, starting with any derivation with no premises, we can go through the same line of reasoning. Adding an arbitrary, unused premise shows us that such a derivation proves all its conclusions to be logical truths. Since we can always modify a premiseless derivation in this way, a premiseless derivation always proves its conclusions to be logical truths:

> A derivation with no premises shows all its conclusions to be logical truths.

Armed with this fact, we can now use derivations to demonstrate that a given sentence is a logical truth. For example, here is a derivation which shows 'Av~A' to be a logical truth:

```
1 |  | ~(Av~A)     A
  |  |--------
2 |  | ~A&~~A     1, DM
3 |  | ~A         2, &E
4 |  | ~~A        3, &E
5 | Av~A          1-4, RD
```

I devised this derivation by using the reductio strategy. I assumed the negation of what I wanted to prove. I then applied the derived De Morgan and reductio rules. Without these derived rules the derivation would have been a lot of work.

Let's try something a bit more challenging. Let's show that

{[A⊃(B&~C)]&(~BvD)}⊃(A⊃D)

is a logical truth. This is not nearly as bad as it seems if you keep your wits about you and look for the main connective. What is the main connective? The second occurrence of '⊃', just after the '}'. Since we want to derive a conditional with '[A⊃(B&~C)]&(~BvD)' as antecedent and 'A⊃D' as consequence, we want a subderivation with the first sentence as assumption and the second as final conclusion:

```
1 |  | [A⊃(B&~C)]&(~BvD)    A
  |  |------------------
  |  |
  |  | ?
  |  |
  |  | ?
  |  |
  |  | A⊃D
  | {[A⊃(B&~C)]&(~BvD)}⊃(A⊃D)    ⊃I
```

What do we do next? Work in from both ends of the subderivation. The conclusion we want is the conditional with 'A' as antecedent. So probably we will want to start a sub-sub-derivation with 'A' as assumption. At the top, our assumption has an '&' as its main connective. So &E will apply to give us two simpler conjuncts which we may be able to use. The first of these conjuncts is a conditional with 'A' as antecedent. We are going to be assuming 'A' as a new assumption any way, so most likely we will be able to apply ⊃E. Let's write down what we have so far:


```
1 |  | [A⊃(B&~C)]&(~BvD)       A
  |  |------------------
2 |  |  | A                    A
  |  |  |---
3 |  |  | [A⊃(B&~C)]&(~BvD)    1, R
4 |  |  | A⊃(B&~C)             3, &E
5 |  |  | ~BvD                 3, &E
6 |  |  |
  |  |  | ?
  |  |  |
  |  |  | D
  |  | A⊃D                     ⊃I
  | {[A⊃(B&~C)]&(~BvD)}⊃(A⊃D)  ⊃I
```

To complete the derivation, we note that from lines 2 and 4 we can get the conjunction 'B&~C' by ⊃E. We can then extract 'B' from 'B&~C by &E and apply the derived form of vE to 'B' and '~BvD' to get 'D' as we needed:

```
1  |  | [A⊃(B&~C)]&(~BvD)       A
   |  |------------------
2  |  |  | A                    A
   |  |  |---
3  |  |  | [A⊃(B&~C)]&(~BvD)    1, R
4  |  |  | A⊃(B&~C)             3, &E
5  |  |  | ~BvD                 3, &E
6  |  |  | B&~C                 2, 4, ⊃E
7  |  |  | B                    6, &E
8  |  |  | D                    5, 7, vE
9  |  | A⊃D                     2-8, ⊃I
10 | {[A⊃(B&~C)]&(~BvD)}⊃(A⊃D)  1-9, ⊃I
```

You might be entertained to know how I dreamed up this horrible-looking example. Note that if, in the last derivation, we eliminated line 10 and the outermost scope line, line 1 would become the premise of a derivation with 'A⊃D' as its final conclusion. In other words, I would have a derivation that in outline looked like this:

```
| X   P
|----
| .
| .
| .
|
| Y   final conclusion
```

But starting with such a derivation I can obviously do the reverse. I get back to the former derivation if I add back the extra outer scope line, call what was the premise the assumption of the subderivation, and add as a last step an application of ⊃I. In outline, I have

```
|  | X     A
|  |---
|  |  .
|  |  .
|  |  .
|  | Y
|  X⊃Y     ⊃I
```

Looking at the last two schematic diagrams you can see that whenever you have a derivation in the form of one, you can easily rewrite it to make it look like the other. This corresponds to something logicians call the *Deduction Theorem.*

Here is one last application. Recall from chapter 3 that a contradiction is a sentence which is false for every assignment of truth values to sentence letters. We can also use derivations to establish that a sentence is a contradiction. Before reading on, see if you can figure out how to do this.

A sentence is a contradiction if and only if it is false in every case. But a sentence is false in every case if and only if its **negation** is true in every case. So all we have to do is to show the negation of our sentence to be a logical truth:

> To demonstrate a sentence, **X**, to be a contradiction, demonstrate its negation, ~**X**, to be a logical truth. That is, construct a derivation with no premises, with ~**X** as the final conclusion.

EXERCISES

7–5. Demonstrate the correctness of the following alternative test for contradictions:

> A derivation with a sentence, **X** as its only premise and two sentences, **Y** and ~**Y**, as conclusions shows **X** to be a contradiction.

7–6. Provide derivations which establish that the following sentences are logical truths. Use derived as well as primitive rules.

a) (AvB)⊃(~B⊃A)
b) Mv~(M&N)
c) [H⊃(O⊃N)]⊃[(H&O)⊃N]
d) (D⊃B)⊃{(D⊃T)⊃[D⊃(B&T)]}
e) (K⊃F)⊃[~F⊃~(K&P)]
f) [(FvG)⊃(P&Q)]⊃(~Q⊃~F)
g) [L⊃(M⊃N)]⊃[(L⊃M)⊃(L⊃N)]
h) [(SvT)⊃F]⊃{[(FvG)⊃H]⊃(S⊃H)}
i) (I&~J)v[(J&K)v~(K&I)]
j) {[C&(AvD)]v~(C&F)}v~(A&~G)

7–7. Provide derivations which establish that the following sentences are contradictions:

a) A&~A
b) (Hv~B)&[(~B⊃H)&~H]
c) [(H&F)⊃C]&~[H⊃(F⊃C)]
d) [~(GvQ)&(K⊃G)]&~(Pv~K)]
e) [K⊃(D⊃P)]&[(~KvD)&~(K⊃P)]
f) ~[~(Nv~R)⊃(N≡~R)]
g) (FvG)≡(~F&~G)
h) [~(FvG)v(P&Q)]&~(~Q⊃~F)
i) (A⊃D)&[(A&~B)v(A&~C)]&[(B&~D)v(B&C)]

(Exercise i is unreasonably long unless you use a derived rule for the distributive law. You have really done the work for proving this law in problem 7–1d.

j) (A≡B)≡(~A≡B)

7–8. Consider the definition

> A set of sentence logic sentences is *Inconsistent* if and only if there is no assignment of truth values to sentence letters which makes all of the sentences in the set true.

a) Explain the connection between inconsistency as just defined and what it is for a sentence to be a contradiction.

b) Devise a way of using derivations to show that a set of sentences is inconsistent.

c) Use your test to establish the inconsistency of the following sets of sentences:

c1) C≡G, G≡~C

c2) FvT, (FvT)⊃(~F&~T)

c3) $J \vee K$, $\sim J \vee \sim K$, $J \equiv K$

c4) $(G \vee K) \supset A$, $(A \vee H) \supset G$, $G \& \sim A$

c5) $D \equiv (\sim P \& \sim M)$, $P \equiv (J \& \sim F)$, $\sim F \vee \sim D$, $D \& J$

7–9. Devise a way of using derivations which will apply to two logically equivalent sentences to show that they are logically equivalent. Explain why your method works. Try your method out on some logical equivalences taken from the text and problems of chapter 3.

CHAPTER SUMMARY EXERCISES

Provide short explanations for each of the following. Check against the text to make sure your explanations are correct, and save your answers for reference and review.

a) Main Connective
b) Primitive Rule
c) Derived Rule
d) Weakening Rule
e) Contraposition Rule
f) De Morgan's Rules
g) Conditional Rules
h) Reductio Ad Absurdum Rule
i) Derivations without Premises
j) Tests for Logical Truths and Contradictions