Working with Multiple Assumptions

As we noted at the end of "Working with Negation Introduction and Negation Elimination", there is no intrinsic reason to confine ourselves to one assumption. Indeed, to complete certain derivations, we will need multiple assumptions. A given derivation may contain multiple assumptions in one of two ways. In the simpler case we may make one assumption, terminate and discharge it, and then at some subsequent point introduce another subderivation to obtain some other sentence we need. You should have no problem with such derivations. They do not involve anything "new". Here is an example of where such multiple assumptions are necessary. We will show that B & D |- (A -> B) & (C -> D):

1. B & D	Premise
2. B	1 &E
3. D	1 &E
4. A	* A
5. B	* 2 Reit
6. A -> B	4-5 ->I
7. C	* A
8. D	* 3 Reit
9. C -> D	7-8 ->I
10. (A -> B) & (C -> D)	6, 9 &I

Note that lines 4 and 5 are inaccessible at all points below 5.

Typically we derive biconditionals by obtaining the two conditionals as lines, using &I and then <->I. Here is an example of such a derivation where we show that ~B v A, ~B -> ~~B |- A <-> B. In this derivation we will take advantage of a point exhibited in the preceding example. If we are given q we can derive p -> q. Consequently in order to get A -> B we will first show that B:

1.	~B v A	Premise
2.	~B -> ~~B	Premise
3.	~B	*A
4.	~~B	* 2, 3 ->E
5.	~ B & ~~B	* 3, 4 &I
6.	B	3-5 ~E
7.	A	* A
8.	B	* 6 Reit
9.	A -> B	7-8 ->I
10.	~~B	6 ~~I
11.	A	1, 10 DS
12.	B	* A
13.	A	* 11 Reit
14.	B -> A	12-13 ->I
15.	(A -> B) & (B -> A)	9, 14 &I
16.	A <-> B	15 <->I

This is rather long, but introduces nothing new. One point to remember, since it will often come in handy, is that if you have a line ~p -> ~~p you can always get p. Here this is illustrated in lines 2 through 6. With basically the same structure you can get ~p if you have p -> ~p and p if you have ~p -> p.

In the other kind of case, you have assumptions within assumptions. Recall that an assumption can be introduced at any point. Regardless of when we introduce one, we must indent one more column. We could, for example, have shown that ~B v A, ~B -> ~~B |- A <-> B in the following way (initially we look only at the first few steps):

1.	~B v A		Premise
2.	~B -> ~~B		Premise
3.	A		A
4.		~B	*A
5.		~~B	* 2, 4 ->E
6.		~B & ~~B	* 4, 5 &I
7.	B		4-6 ~E

Notice that at this point the assumption we made at line 4 is discharged. Here I used ~E rather than ~I at line 8 since I noticed that I would want to use ->I. When you discharge an assumption, be very careful to move only one column to the left. These lines are still under the assumption we made at line 3. Here are the next stages of this derivation:

1.	~B v A			Premise
2.	~B -> ~~B			Premise
3.		A		A
4.			~B	*A
5.			~~B	* 2, 4 ->E
6.			~B & ~~B	* 4, 5 &I
7.		B		4-6 ~E
8.	A -> B			3-7 ->I

Note that at line 8 we moved to the main column, and all of lines 3 through 7 are now inaccessible. If we make another assumption it will indent only one column. Here is the completed derivation:

1.	~B v A			Premise
2.	~B -> ~~B			Premise
3.		A		A
4.			~B	*A
5.			~~B	* 2, 4 ->E
6.			~B & ~~B	* 4, 5 &I
7.		B		4-6 ~E
8.	A -> B			3-7 ->I
9.		B		* A
10.		~~B		*9 ~~I
11.		A		* 1, 10 DS
12.	B -> A			9-11 ->I
13.	(A -> B) & (B -> A)			8, 12 &I
14.	A <-> B			13 <->I

When terminating and discharging you must be careful to discharge only the “nearest” assumption. If you have been indenting correctly, this will be the one in the column in which you have been working. The following is a mistake:

1.	~B v A			Premise
2.	~B -> ~~B			Premise
3.		A		* A
4.			~B	* A
5.			~~B	* 2, 4 ->E
6.			~B & ~~B	* 4, 5 &I
7.		~A		* 3-6 ~I ERRONEOUS

As is this:

1.	~Q			Premise
2.		P		*A
3.			Q	*A
4.			Q v R	* 3 vI
5.			R	* 1, 4 DS
6.		P -> R		2-5 ->I ERRONEOUS

And when you discharge, make sure you move only one column to the left:

1.	~B v A			Premise
2.	~B -> ~~B			Premise
3.		A		* A
4.			~B	* A
5.			~~B	* 2, 4 ->E
6.			~B & ~~B	* 4, 5 &I
7.	~~B			4-6 ~I ERRONEOUS

Let us now look at a few more examples of derivations. Let us show that ~(P & Q) |- ~P v ~Q. We will try to show this by setting up a ~E as follows:

1. ~(P & Q)		Premise
2.	~(~P v ~Q)	A

But what do we do next? Well, we do know that we need a contradiction. Here we have only one premise, and we have no rule that allows us to do anything to it. So it looks like our best hope is to obtain P & Q. Since this is a conjunction we will try to get each of the conjuncts in the same column and then use &I. Once you try this, the derivation is fairly easy:

1. ~(P & Q)		Premise
2.	~(~P v ~Q)	* A
3.	~P	* A
4.	~P v ~Q	* 3 vI
5.	~(~P v ~Q) & (~P v ~Q)	* 2, 4 &I
6.	P	* 3-5 ~E
7.	~Q	* A
8.	~P v ~Q	* 7 vI
9.	~(~P v ~Q) & (~P v ~Q)	* 2, 8 &I
10.	Q	* 7-9 ~E
11.	P & Q	* 6, 10 &I
12.	(P & Q) & ~(P & Q)	* 1, 11 &I
13. ~P v ~Q		2-12 ~E