Working with Negation Introduction and Negation Elimination

We provided in a preceding section a brief discussion of our ~I and ~E rules. Here we will examine them in more detail. Schematically we can exhibit them as follows:

~I

	.	
	.	
m	p	* A
	.	
	.	
m + n	q & ~q	*
m + n + 1	~p	m - m+n ~I

~E

	.	
	.	
m	~p	* A
	.	
	.	
m + n	q & ~q	*
m + n + 1	p	m - m+n ~E

Remember to indent one column whenever you introduce an assumption. And be sure to move one column to the left of the assumption that is discharged when you discharge an assumption.

We cannot here undertake a formal proof of the soundness of ~I, that is, a proof that it will not enable us to derive sentences that do not follow from the premises we have. But in the introduction we provided a sketch of such a proof. We suppose that p is true. We then show that, given the premises, q & ~q will be true. But q & ~q cannot be true. So, given the premises, p cannot be true. Similar reasoning accounts for ~E. Since ~p cannot be true, p must be.

Here is a very trivial example of ~I where we show that ~A |- ~(A & B):

1. ~A		Premise
2.	A & B	* A
3.	A	* 2 &E
4.	A & ~A	* 1, 3 &I
5. ~(A & B)		2-4 ~I

Suppose you wish to show that A & ~B |- ~(A -> B):

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

We mentioned in the introduction that if the sentence you want is not a negation you can use ~E. Supposing you want some sentence p you assume ~p. Here is an easy example where we show that ~(~A v B) |- A.

1. ~(~A v B)		Premise
2.	~A	* A
3.	~A v B	* 2 vI
4.	(~A v B) & ~(~A v B)	* 1,3 &I
5. A		2-4 ~E

How do you know when to set up a ~I or a ~E? While we will discuss this at various points along the way, the best advice now is to set one up in cases where the sentence you need is not a conditional (use an ->I if you wish to obtain a conditional), is not a conjunction (here try to establish the conjuncts separately and then use &I), and is not a biconditional (here try to get separately each of the conditionals you need). But only set it up after you have made any obvious moves that utilize the basic rules. This will sometimes lead to longer derivations, but at least you will know what you have to work with.

We noted in connection with ->I that you may need to use one in order to forward a derivation of something else. The same point obtains here: that is, you may need to use ~I even if it does not immediately yield the conclusion you are seeking. Here is an example:

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

Here we have shown that ~(A <-> B) -> R, ~A & B |- R.

Let us look at one more example before moving on, showing that ~(~A v B), A -> R |- R v S. Note that if we get R we can get the conclusion we want, and that we can get R if we can get A.

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

Up to now we have worked with derivations that incorporate only one subderivation. But nothing in our rules forbids us from introducing as many assumptions as we might want or need. So let us turn to a discussion of multiple assumptions.