Contents

Chapter 1

Proofs in Sentential Logic

1.1   Substitution Instances and Proofs

In the last chapter, we introduced the truth table method, which allowed us to check to see if various logical properties were present, such as whether a statement is a tautology or whether an argument is valid. The method in that chapter was semantic, because it relied on the meaning of symbols, specifically, whether they were interpreted as true or false. The nice thing about that method was that it was completely mechanical. If you just followed the rules like a robot, you would eventually get the right answer. You didn't need any special insight and there were no tough decisions to make. The downside to this method was that the tables quickly became way too long. It just isn't practical to make a 32 line table every time you have to deal with five different sentence letters.

In this chapter, we are going to introduce a new method for checking for validity and other logical properties. This time our method is going to be purely syntactic. We won't be at all concerned with what our symbols mean. We are just going to look at the way they are arranged. Our method here will be called a system of natural deduction. When you use a system of natural deduction, you won't do it mechanically. You will need to understand the logical structure of the argument and employ your insight. This is actually one of the reasons people like systems of natural deduction. They let us represent the logical structure of arguments in a way we can understand. Learning to represent and manipulate arguments this way is a core mental skill, used in fields like mathematics and computer programming.

Consider two arguments in SL:

Argument A   **Argument B**

P_1: $P \lor Q$ P_1: $P \to Q$
P_2: $\neg P$ P_2: P
$\overline{}$ $\overline{}$
C: Q   C: Q

These are both valid arguments. Go ahead and prove that for yourself by constructing the four-line truth tables. These particular valid arguments are examples of important kinds of arguments that are given special names. Argument A is an example of a kind of argument *disjunction elimination* ($\lor$-E). Given a disjunction and the negation of one of the disjuncts, the other disjunct follows as a valid consequence. Argument B makes use of a different valid form: Given a conditional and its antecedent, the consequent follows as a valid consequence. In our system it will be called *conditional elimination* ($\to$-E).

Both of the arguments above remain valid even if we substitute different sentence letters. You don't even need to run the truth tables again to see that these arguments are valid:

Argument A*   **Argument B***

P_1: $A \lor B$ P_1: $A \to B$
P_2: $\neg A$ P_2: A
$\overline{}$ $\overline{}$
C: B   C: B

Replacing P with A and Q with B changes nothing (so long as we are sure to replace *every* P with an A and every Q with a B). What's more interesting is that we can replace the individual sentence letters in Argument A and Argument B with longer sentences in SL and the arguments will still be valid, as long as we do the substitutions consistently. Here are two more perfectly valid instances of disjunction and conditional elimination.

Argument A**   **Argument B****

P_1: $(C \land D) \lor (E \lor F)$ P_1: $(G \to H) \to (I \lor J)$
P_2: $\neg (C \land D)$ P_2: $(G \to H)$
$\overline{}$ $\overline{}$
C: $E \lor F$ C: $I \lor J$

Again, you can check these using truth tables, although the 16 line truth tables begin to get tiresome. All of these arguments are what we call *substitution instances* of the same two logical forms. We call them that because you get them by replacing the sentence letters with other sentences, either sentence letters or longer sentences in SL. A substitution instance cannot change

the sentential connectives of a sentence, however. The sentential connectives are what make the *logical form* of the sentence. We can write these logical forms using fancy script letters.

Disjunction Elimination	**Conditional Elimination**
P_1: $\mathcal{A} \vee \mathcal{B}$	P_1: $\mathcal{A} \rightarrow \mathcal{B}$
P_2: $\neg\mathcal{A}$	P_2: $\mathcal{A}$
C: $\mathcal{B}$	C: $\mathcal{B}$

As we explained earlier, the fancy script letters are *metavariables*. They are a part of our metalanguage and can refer to single sentence letters like P or longer sentences like $A \leftrightarrow (B \wedge (C \vee D))$.

Formally, we can define a SENTENCE FORM as a sentence in SL that contains one or more metavariables in place of sentence letters. A SUBSTITUTION INSTANCE of that sentence form is then a sentence created by consistently substituting sentences for one or more of the metavariables in the sentence form. Here "consistently substituting" means replacing all instances of the metavariable with the same sentence. You cannot replace instances of the same metavariable with different sentences, or leave a metavariable as it is, if you have replaced other metavariables of that same type. An ARGUMENT FORM is an argument that includes one or more sentence forms, and a SUBSTITUTION INSTANCE of the argument form is the argument obtained by consistently replacing the sentence forms in the argument form with their substitution instances.

Once we start identifying valid argument forms like this, we have a new way of showing that longer arguments are valid. Truth tables are fun, but doing the 1028 line truth table for an argument with 10 sentence letters would be tedious. Worse, we would never be sure we hadn't made a little mistake in all those Ts and Fs. Part of the problem is that we have no way of knowing *why* the argument is valid. The table gives you very little insight into how the premises work together.

The aim of a *proof system* is to show that particular arguments are valid in a way that allows us to understand the reasoning involved in the argument. Instead of representing all the premises and the conclusion in one table, we break the argument up into steps. Each step is a basic argument form of the sort we saw above, like disjunctive syllogism or modus ponens. Suppose we are given the premises $\neg L \rightarrow (J \vee L)$ and $\neg L$ and wanted to show J. We can break this up into two smaller arguments, each of which is a substitution inference of a form we know is correct.

Argument 1	**Argument 2**
P_1: $\neg L \rightarrow (J \vee L)$	P_1: $J \vee L$
P_2: $\neg L$	P_2: $\neg L$
C: $J \vee L$	C: J

The first argument is a substitution instance of modus ponens and the second is a substitution instance of disjunctive syllogism, so we know they are both valid. Notice also that the conclusion of the first argument is the first premise of the second, and the second premise is the same in both arguments. Together, these arguments are enough to get us from $\neg L \to (J \vee L)$ and $\neg L$ to J.

These two arguments take up a lot of space, though. To complete our proof system, we need a system for showing clearly how simple steps can combine to get us from premises to conclusions. The system we will use in this book was devised by the American logician Frederic Brenton Fitch (1908–1987). We begin by writing our premises on numbered lines with a bar on the left and a little bar underneath to represent the end of the premises. Then we write "Want" on the side followed by the conclusion we are trying to reach. If we wanted to write out arguments 1 and 2 above, we would begin like this.

$$
\begin{array}{r|ll}
1 & \neg L \to (J \vee\ L) & \\
2 & \neg L & \text{Want: } J \\
\end{array}
$$

We then add the steps leading to the conclusion below the horizontal line, each time explaining off to the right why we are allowed to write the new line. This explanation consists of citing a rule and the prior lines the rule is applied to. In the example we have been working with we would begin like this

$$
\begin{array}{r|ll}
1 & \neg L \to (J \vee L) & \\
2 & \neg L & \text{Want: } J \\
3 & J \vee L & \to\text{E } 1,\ 2 \\
\end{array}
$$

and then go like this

$$
\begin{array}{r|ll}
1 & \neg L \to (J \vee L) & \\
2 & \neg L & \text{Want: } J \\
3 & J \vee L & \to\text{E } 1,\ 2 \\
4 & J & \vee\text{E } 2,\ 3 \\
\end{array}
$$

The little chart above is a *proof* that J follows from $\neg L \to (J \vee L)$ and $\neg L$. We will also call proofs like this *derivations*. Formally, a PROOF is a sequence of sentences. The first sentences of the sequence are assumptions; these are the premises of the argument. Every sentence later in the sequence follows from earlier sentences by one of the rules of proof. The final sentence of the sequence is the conclusion of the argument.

1.2 Basic Rules for Sentential Logic

In designing a proof system, we could just start with disjunctive syllogism and modus ponens. Whenever we discovered a valid argument that could not be proved with rules we already had, we could introduce new rules. Proceeding in this way, we would have an unsystematic grab bag of rules. We might accidentally add some strange rules, and we would surely end up with more rules than we need.

Instead, we will develop what is called a SYSTEM OF NATURAL DEDUCTION. In a natural deduction system, there will be two rules for each logical operator: an introduction, and an elimination rule. The introduction rule will allow us to prove a sentence that has the operator you are "introducing" as its main connective. The elimination rule will allow us to prove something given a sentence that has the operator we are "eliminating" as the main logical operator.

In addition to the rules for each logical operator, we will also have a reiteration rule. If you already have shown something in the course of a proof, the reiteration rule allows you to repeat it on a new line. We can define the RULE OF REITERATION like this

Reiteration (R)

$$
\begin{array}{c|ll}
m & \mathcal{A} & \\
n & \mathcal{A} & \text{R } m
\end{array}
$$

This diagram shows how you can add lines to a proof using the rule of reiteration. As before, the script letters represent sentences of any length. The upper line shows the sentence that comes earlier in the proof, and the bottom line shows the new sentence you are allowed to write and how you justify it. The reiteration rule above is justified by one line, the line that you are reiterating. So the "R m" on line 2 of the proof means that the line is justified by the reiteration rule (R) applied to line m. The letters m and n are variables, not real line numbers. In a real proof, they might be lines 5 and 7, or lines 1 and 2, or whatever. When we define the rule, however, we use variables to underscore the point that the rule may be applied to any line that is already in the proof.

Obviously, the reiteration rule will not allow us to show anything *new*. For that, we will need more rules. The remainder of this section will give six basic introduction and elimination rules. This will be enough to do some basic proofs in SL. Sections 4.3 through 4.5 will explain introduction rules involved in fancier kinds of derivation called conditional proof and indirect proof. The remaining sections of this chapter will develop our system of natural deduction further and give you tips for playing in it.

All of the rules introduced in this chapter are summarized starting on p.**??**.

Conjunction

Think for a moment: What would you need to show in order to prove $E \wedge F$?

Of course, you could show $E \wedge F$ by proving E and separately proving F. This holds even if the two conjuncts are not atomic sentences. If you can prove $[(A \vee J) \to V]$ and $[(V \to L) \leftrightarrow (F \vee N)]$, then you have effectively proved $[(A \vee J) \to V] \wedge [(V \to L) \leftrightarrow (F \vee N)]$. So this will be our conjunction introduction rule, which we abbreviate $\wedge$I:

$$
\begin{array}{ll}
m & \mathcal{A} \\
n & \mathcal{B} \\
  & \mathcal{A} \wedge \mathcal{B} \qquad \wedge\text{I } m,\, n
\end{array}
\qquad\qquad
\begin{array}{ll}
m & \mathcal{A} \\
n & \mathcal{B} \\
  & \mathcal{B} \wedge \mathcal{A} \qquad \wedge\text{I } m,\, n
\end{array}
$$

A line of proof must be justified by some rule, and here we have "$\wedge$I m, n." This means: Conjunction introduction applied to line m and line n. Again, these are variables, not real line numbers; m is some line and n is some other line. If you have K on line 8 and L on line 15, you can prove $(K \wedge L)$ at some later point in the proof with the justification "$\wedge$I 8, 15."

We have written two versions of the rule to indicate that you can write the conjuncts in any order. Even though K occurs before L in the proof, you can derive $(L \wedge K)$ from them using the right-hand version $\wedge$I. You do not need to mark this in any special way in the proof.

Now, consider the elimination rule for conjunction. What are you entitled to conclude from a sentence like $E \wedge F$? Surely, you are entitled to conclude E; if $E \wedge F$ were true, then E would be true. Similarly, you are entitled to conclude F. This will be our conjunction elimination rule, which we abbreviate $\wedge$E:

$$
\begin{array}{ll}
m & \mathcal{A} \wedge \mathcal{B} \\
  & \mathcal{A} \qquad \wedge\text{E } m
\end{array}
\qquad\qquad
\begin{array}{ll}
m & \mathcal{A} \wedge \mathcal{B} \\
  & \mathcal{B} \qquad \wedge\text{E } m
\end{array}
$$

When you have a conjunction on some line of a proof, you can use $\wedge$E to derive either of the conjuncts. Again, we have written two versions of the rule to indicate that it can be applied to either side of the conjunction. The $\wedge$E rule requires only one sentence, so we write one line number as the justification for applying it. For example, both of these moves are acceptable in derivations.

$$
\begin{array}{ll}
4 & A \wedge (B \vee C) \\
5 & A \qquad\qquad \wedge\text{E } 4
\end{array}
\qquad\qquad
\begin{array}{ll}
10 & A \wedge (B \vee C) \\
\ldots & \ldots \\
15 & (B \vee C) \qquad \wedge\text{E } 10
\end{array}
$$

Some textbooks will only let you use $\wedge$E on one side of a conjunction. They then make you *prove* that it works for the other side. We won't do this, because it is a pain in the neck.

Even with just these two rules, we can provide some proofs. Consider this argument.

$$[(A \lor B) \to (C \lor D)] \land [(E \lor F) \to (G \lor H)]$$
$$\therefore \ [(E \lor F) \to (G \lor H)] \land [(A \lor B) \to (C \lor D)]$$

The main logical operator in both the premise and conclusion is a conjunction. Since the conjunction is symmetric, the argument is obviously valid. In order to provide a proof, we begin by writing down the premise. After the premises, we draw a horizontal line—everything below this line must be justified by a rule of proof. So the beginning of the proof looks like this:

1 | $[(A \lor B) \to (C \lor D)] \land [(E \lor F) \to (G \lor H)]$

From the premise, we can get each of the conjuncts by $\land$E. The proof now looks like this:

1	$[(A \lor B) \to (C \lor D)] \land [(E \lor F) \to (G \lor H)]$	
2	$[(A \lor B) \to (C \lor D)]$	$\land$E 1
3	$[(E \lor F) \to (G \lor H)]$	$\land$E 1

The rule $\land$I requires that we have each of the conjuncts available somewhere in the proof. They can be separated from one another, and they can appear in any order. So by applying the $\land$I rule to lines 3 and 2, we arrive at the desired conclusion. The finished proof looks like this:

1	$[(A \lor B) \to (C \lor D)] \land [(E \lor F) \to (G \lor H)]$	
2	$[(A \lor B) \to (C \lor D)]$	$\land$E 1
3	$[(E \lor F) \to (G \lor H)]$	$\land$E 1
4	$[(E \lor F) \to (G \lor H)] \land [(A \lor B) \to (C \lor D)]$	$\land$I 3, 2

This proof is trivial, but it shows how we can use rules of proof together to demonstrate the validity of an argument form. Also: Using a truth table to show that this argument is valid would have required a staggering 256 lines, since there are eight sentence letters in the argument.

Disjunction

If M were true, then $M \lor N$ would also be true. So the disjunction introduction rule ($\lor$I) allows us to derive a disjunction if we have one of the two disjuncts:

m	$\mathcal{A}$			m	$\mathcal{A}$	
	$\mathcal{A} \lor \mathcal{B}$	$\lor$I m			$\mathcal{B} \lor \mathcal{A}$	$\lor$I m

Like the rule of conjunction elimination, this rule can be applied two ways. Also notice that $\mathcal{B}$ can be *any* sentence whatsoever. So the following is a legitimate proof:

$$
\begin{array}{l|l}
1 & M \\ \hline
2 & M \vee ([(A \leftrightarrow B) \rightarrow (C \wedge D)] \leftrightarrow [E \wedge F]) \qquad \vee\text{I } 1
\end{array}
$$

This might seem odd. How can we prove a sentence that includes A, B, and the rest, from the simple sentence M—which has nothing to do with the other letters? The secret here is to remember that all the new letters are on just one side of a disjunction, and nothing on that side of the disjunction has to be true. As long as M is true, we can add whatever we want after a disjunction and the whole thing will continue to be true.

Now consider the disjunction elimination rule. What can you conclude from $M \vee N$? You cannot conclude M. It might be M's truth that makes $M \vee N$ true, as in the example above, but it might not. From $M \vee N$ alone, you cannot conclude anything about either M or N specifically. If you also knew that N was false, however, then you would be able to conclude M.

$$
\begin{array}{l|l} \qquad\qquad\qquad\qquad
m & \mathcal{A} \vee \mathcal{B} \\
n & \neg \mathcal{B} \\
  & \mathcal{A} \qquad\quad \vee\text{E } m, n
\end{array}
\qquad\qquad
\begin{array}{l|l}
m & \mathcal{A} \vee \mathcal{B} \\
n & \neg \mathcal{A} \\
  & \mathcal{B} \qquad\quad \vee\text{E } m, n
\end{array}
$$

We've seen this rule before: it is just disjunctive syllogism. Now that we are using a system of natural deduction, we are going to make it our rule for disjunction elimination ($\vee$E). Once again, the rule works on both sides of the sentential connective.

Conditionals and biconditionals

The rule for conditional introduction is complicated because it requires a whole new kind of proof, called conditional proof. We will deal with this in the next section. For now, we will only use the rule of conditional elimination.

Nothing follows from $M \rightarrow N$ alone, but if we have both $M \rightarrow N$ and M, then we can conclude N. This is another rule we've seen before: modus ponens. It now enters our system of natural deduction as the conditional elimination rule ($\rightarrow$E).

$$
\begin{array}{l|l}
m & \mathcal{A} \rightarrow \mathcal{B} \\
n & \mathcal{A} \\
  & \mathcal{B} \qquad\quad \rightarrow\text{E } m, n
\end{array}
$$

Biconditional elimination ($\leftrightarrow$E) will be a double-barreled version of conditional elimination. If you have the left-hand subsentence of the biconditional, you can derive the right-hand

subsentence. If you have the right-hand subsentence, you can derive the left-hand subsentence. This is the rule:

m	$\mathcal{A} \leftrightarrow \mathcal{B}$		m	$\mathcal{A} \leftrightarrow \mathcal{B}$	
n	$\mathcal{A}$		n	$\mathcal{B}$	
	$\mathcal{B}$	$\leftrightarrow$E m, n		$\mathcal{A}$	$\leftrightarrow$E m, n

Invalid argument forms

In section 4.1, in the last two problem parts (p. **??**), we saw that sometimes an argument looks like a legitimate substitution instance of a valid argument form, but really isn't. For instance, the problem set C asked you to identify instances of modus ponens. Below I'm giving you two of the answers.

(5) Modus ponens

1. $(C \wedge L) \rightarrow (E \vee C)$
2. $C \wedge L$

∴ $E \vee C$

(7) *Not* modus ponens.

1. $D \rightarrow E$
2. E

∴ D

The argument on the left is an example of a valid argument, because it is an instance of modus ponens, while the argument on the right is an example of an invalid argument, because it is not an example of modus ponens. (We originally defined the terms valid and invalid on p. **??**). Arguments like the one on the right, which try to trick you into thinking that they are instances of valid arguments, are called DEDUCTIVE FALLACIES. The argument on the right is specifically called the fallacy of AFFIRMING THE CONSEQUENT. In the system of natural deduction we are using in this textbook, modus ponens has been renamed "conditional elimination," but it still works the same way. So you will need to be on the lookout for deductive fallacies like affirming the consequent as you construct proofs.

Notation

The rules we have learned in this chapter give us enough to start doing some basic derivations in SL. This will allow us to prove things syntactically which would have been too cumbersome to prove using the semantic method of truth tables. We now need to introduce a few more symbols to be clear about what methods of proof we are using.

In Chapter 1, we used the three dots ∴ to indicate generally that one thing followed from another. In chapter 3 we introduced the double turnstile, $\models$, to indicate that one statement could

be proven some others using truth tables. Now we are going to use a single turnstile, $\vdash$, to indicate that we can derive a statement from a bunch of premises, using the system of natural deduction we have begun to introduce in this section. Thus we will write $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\} \vdash \mathcal{D}$, to indicate that there is a derivation going from the premises $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ to the conclusion $\mathcal{D}$. Note that these are metavariables, so I could be talking about any sentences in SL.

The single turnstile will work the same way the double turnstile did. So, in addition to the uses of the single turnstile above we can write $\vdash \mathcal{A}$ to indicate that $\mathcal{A}$ can be proven a tautology using syntactic methods. We can write $\mathcal{A} \dashv\vdash \mathcal{B}$ to say that $\mathcal{A}$ and $\mathcal{B}$ can be proven logically equivalent using these derivations. You will learn how to do these later things at the end of the chapter. In the meantime, we need to practice our basic rules of derivation.

1.3   Conditional Proof

So far we have introduced introduction and elimination rules for the conjunction and disjunction, and elimination rules for the conditional and biconditional, but we have no introduction rules for conditionals and biconditionals, and no rules at all for negations. That's because these other rules require fancy kinds of derivations that involve putting proofs inside proofs. In this section, we will look at one of these kinds of proof, called conditional proof.

Conditional introduction

Consider this argument:

   1. $R \vee F$
          ―――――――――――
  $\therefore\ \neg R \to F$

The argument is valid. You can use the truth table to check it. Unfortunately, we don't have a way to prove it in our syntactic system of derivation. To help us see what our rule for conditional introduction should be, we can try to figure out what new rule would let us prove this obviously true argument.

Let's start the proof in the usual way, like this:

   1 $\left|\ R \vee F\right.$ Want: ¬R →F

If we had $\neg R$ as a further premise, we could derive F by the $\vee$E rule. But sadly, we do not have $\neg R$ as a premise, and we can't derive it directly from the premise we do have—so we cannot simply prove F. What we will do instead is start a *subproof*, a proof within the main proof. When we start a subproof, we draw another vertical line to indicate that we are no longer in the main

proof. Then we write in an assumption for the subproof. This can be anything we want. Here, it will be helpful to assume $\neg R$. Our proof now looks like this:

$$
\begin{array}{l|l}
1 & R \vee F \qquad \text{Want: } \neg R \rightarrow F \\
2 & \quad \boxed{\neg R} \qquad \text{Assumption for CD, Want: F}
\end{array}
$$

It is important to notice that we are not claiming to have proved $\neg R$. We do not need to write in any justification for the assumption line of a subproof. You can think of the subproof as posing the question: What could we show *if* $\neg R$ were true? For one thing, we can derive F. To make this completely clear, I have annotated line 2 "Assumption for CD," to indicate that this is an additional assumption we are making because we are using conditional derivation (CD). I have also added "Want: F" because that is what we will want to show during the subderivation. In the future I won't always include all this information in the annotation. But for now we will use it to be completely clear on what we will be doing.

So now let's go ahead and show F in the subderivation.

$$
\begin{array}{l|l}
1 & R \vee F \qquad \text{Want: } \neg R \rightarrow F \\
2 & \quad \neg R \qquad \text{Assumption for CD, Want: F} \\
3 & \quad F \qquad \vee\text{E 1, 2}
\end{array}
$$

This has shown that *if* we had $\neg R$ as a premise, *then* we could prove F. In effect, we have proven $\neg R \rightarrow F$. So the conditional introduction rule ($\rightarrow$I) will allow us to close the subproof and derive $\neg R \rightarrow F$ in the main proof. Our final proof looks like this:

$$
\begin{array}{l|l}
1 & R \vee F \qquad \text{Want: } \neg R \rightarrow F \\
2 & \quad \neg R \qquad \text{Assumption for CD, Want: F} \\
3 & \quad F \qquad \vee\text{E 1, 2} \\
4 & \neg R \rightarrow F \qquad \rightarrow\text{I 2–3}
\end{array}
$$

Notice that the justification for applying the $\rightarrow$I rule is the entire subproof. Usually that will be more than just two lines.

Now that we have that example, let's lay out more precisely the rules for subproofs and then give the formal schemes for the rule of conditional and biconditional introduction.

RULE 1 You can start a subproof on any line, except the last one, and introduce any assumptions with that subproof.

RULE 2 All subproofs must be closed by the time the proof is over.

RULE 3 Subproofs may closed at any time. Once closed, they can be used to justify →I, ↔I, ¬E, and ¬I.

RULE 4 Nested subproofs must be closed before the outer subproof is closed.

RULE 5 Once the subproof is closed, lines in the subproof cannot be used in later justifications.

Rule 1 gives you great power. You can assume anything you want, at any time. But with great power, comes great responsibility, and rules 2–5 explain what your responsibilities are. Making an assumption creates the burden of starting a subproof, and subproofs must end before the proof is done. (That's why we can't start a subproof on the last line.) Closing a subproof is called *discharging* the assumptions of that subproof. So we can summarize your responsibilities this way: You cannot complete a proof until you have discharged all of the assumptions introduced in subproofs. Once the assumptions are discharged, you can use the whole subproof as a justification, but not the individual lines. So you need to know going into the subproof what you are going to use it for once you get out. As in so many parts of life, you need an exit strategy.

With those rules for subproofs in mind, the →I rule looks like this:

$$
\begin{array}{lll}
m & \quad\mathcal{A} & \text{want }\mathcal{B} \\
n & \quad\mathcal{B} & \\
  & \mathcal{A}\to\mathcal{B} & \to\text{I } m\text{--}n
\end{array}
$$

You still might think this gives us too much power. In logic, the ultimate sign you have too much power is that given any premise $\mathcal{A}$ you can prove any conclusion $\mathcal{B}$. Fortunately, our rules for subproofs don't let us do this. Imagine a proof that looks like this:

$$
\begin{array}{ll}
1 & \mathcal{A} \\
2 & \quad\mathcal{B}
\end{array}
$$

It may seem as if a proof like this will let you reach any conclusion $\mathcal{B}$ from any premise $\mathcal{A}$. But this is not the case. By rule 2, in order to complete a proof, you must close all of the subproofs, and we haven't done that. A subproof is only closed when the vertical line for that subproof ends. To put it another way, you can't end a proof and still have two vertical lines going.

You still might think this system gives you too much power. Maybe we can try closing the subproof and writing $\mathcal{B}$ in the main proof, like this

$$
\begin{array}{lll}
1 & \mathcal{A} & \\
2 & \quad\mathcal{B} & \\
3 & \quad\mathcal{B} & \text{R } 2 \\
4 & \mathcal{B} & \text{R } 3
\end{array}
$$

But this is wrong, too. By rule 5, once you close a subproof, you cannot refer back to individual lines inside it.

Of course, it is legitimate to do this:

1 | $\mathcal{A}$
2 | $\mathcal{B}$
3 | $\mathcal{B}$ R 2
4 | $\mathcal{B} \rightarrow \mathcal{B}$ $\rightarrow$I 2–3

This should not seem so strange, though. Since $\mathcal{B} \rightarrow \mathcal{B}$ is a tautology, no particular premises should be required to validly derive it. (Indeed, as we will see, a tautology follows from any premises.)

When we introduce a subproof, we typically write what we want to derive in the right column, just like we did in the first example in this section. This is just so that we do not forget why we started the subproof if it goes on for five or ten lines. There is no "want" rule. It is a note to ourselves and not formally part of the proof.

Having an exit strategy when you launch a subproof is crucial. Even if you discharge an assumption properly, you might wind up with a final line that doesn't do you any good. In order to derive a conditional by $\rightarrow$I, for instance, you must assume the antecedent of the conditional in a subproof. The last line of the subproof must be the consequent of the conditional, and the whole conditional is the first line after the end of the subproof. Pick your assumptions so that you wind up with a conditional that you actually need. It is always permissible to close a subproof and discharge its assumptions, but it will not be helpful to do so until you get what you want.

Now that we have the rule for conditional introduction, consider this argument:

   1. $P \rightarrow Q$
   2. $Q \rightarrow R$

$\therefore\ P \rightarrow R$

We begin the proof by writing the two premises as assumptions. Since the main logical operator in the conclusion is a conditional, we can expect to use the $\rightarrow$I rule. For that, we need a subproof—so we write in the antecedent of the conditional as an assumption of a subproof:

1 | $P \rightarrow Q$
2 | $Q \rightarrow R$
3 | P

We made P available by assuming it in a subproof, allowing us to use $\rightarrow$E on the first premise. This gives us Q, which allows us to use $\rightarrow$E on the second premise. Having derived R, we close the subproof. By assuming P we were able to prove R, so we apply the $\rightarrow$I rule and finish the proof.

$$
\begin{array}{lll}
1 & P \rightarrow Q & \\
2 & Q \rightarrow R & \\
3 & \quad P & \text{want } R \\
4 & \quad Q & \rightarrow\text{E } 1, 3 \\
5 & \quad R & \rightarrow\text{E } 2, 4 \\
6 & P \rightarrow R & \rightarrow\text{I } 3\text{--}5 \\
\end{array}
$$

Biconditional introduction

Just as the rule for biconditional elimination was a double-headed version of conditional elimination, our rule for biconditional introduction is a double-headed version of conditional introduction. In order to derive $W \leftrightarrow X$, for instance, you must be able to prove X by assuming W *and* prove W by assuming X. The biconditional introduction rule ($\leftrightarrow$I) requires two subproofs. The subproofs can come in any order, and the second subproof does not need to come immediately after the first—but schematically, the rule works like this:

$$
\begin{array}{lll}
m & \quad \mathcal{A} & \text{want } \mathcal{B} \\
n & \quad \mathcal{B} & \\
p & \quad \mathcal{B} & \text{want } \mathcal{A} \\
q & \quad \mathcal{A} & \\
  & \mathcal{A} \leftrightarrow \mathcal{B} & \leftrightarrow\text{I } m\text{--}n, p\text{--}q \\
\end{array}
$$

We will call any proof that uses subproofs and either $\rightarrow$I or $\leftrightarrow$I CONDITIONAL PROOF. By contrast, the first kind of proof you learned, where you only use the six basic rules, will be called DIRECT PROOF. In section 4.3 we will learn the third and final kind of proof *indirect proof*. But for now you should practice conditional proof.

1.4   Indirect Proof

The last two rules we need to discuss are negation introduction ($\neg$I) and negation elimination ($\neg$E). As with the rules of conditional and biconditional introduction, we have put off explaining the rules, because they require launching subproofs. In the case of negation introduction and

elimination, these subproofs are designed to let us perform a special kind of derivation classically known as *reductio ad absurdum*, or simply *reductio*.

A *reductio* in logic is a variation on a tactic we use in ordinary arguments all the time. In arguments we often stop to imagine, for a second, that what our opponent is saying is true, and then realize that it has unacceptable consequences. In so-called "slippery slope" arguments or "arguments from consequences," we claim that doing one thing will will lead us to doing another thing which would be horrible. For instance, you might argue that legalizing physician assisted suicide for some patients might lead to the involuntary termination of lots of other sick people. These arguments are typically not very good, but they have a basic pattern whcih we can make rigorous in our logical system. These arguments say "if my opponent wins, all hell will break loose." In logic the equivalent of all hell breaking loose is asserting a contradiction. The worst thing you can do in logic is contradict yourself. The equivalent of our opponent being right in logic would be that a sentence we are trying to prove true turns out to be false (or alternately, that a sentence we are trying to prove false turns out to be true.) So, in developing the rules for reductio ad absurdum, we need to find a way to say "if this sentence were false (or true), we would have to assert a contradiction."

Here is a simple mathematical argument in English for the conclusion that there is no largest number:

> Assume there *is* some greatest natural number. Call it A.
> That number plus one is also a natural number.
> Obviously, $A + 1 > A$.
> So there is a natural number greater than A.
> This is impossible, since A is assumed to be the greatest natural number.
> ∴ There is no greatest natural number.

This argument form is traditionally called a *reductio*. Its full Latin name is *reductio ad absurdum*, which means "reduction to absurdity." In a reductio, we assume something for the sake of argument—for example, that there is a greatest natural number. Then we show that the assumption leads to two contradictory sentences—for example, that A is the greatest natural number and that it is not. In this way, we show that the original assumption must have been false.

In our system of natural deduction, this kind of proof will be known as INDIRECT PROOF. The basic rules for negation will allow for arguments like this. If we assume something and show that it leads to contradictory sentences, then we have proven the negation of the assumption. This is the negation introduction (¬I) rule:

m		$\mathcal{A}$	for reductio	m		$\mathcal{A}$	for reductio
n		$\mathcal{B}$		n		$\neg\mathcal{B}$	
$n+1$		$\neg\mathcal{B}$		$n+1$		$\mathcal{B}$	
$n+2$	$\neg\mathcal{A}$		$\neg$I m–$n+1$	$n+2$	$\neg\mathcal{A}$		$\neg$I m–$n+1$

For the rule to apply, the last two lines of the subproof must be an explicit contradiction: either the second sentence is the direct negation of the first, or vice versa. We write "for reductio" as a note to ourselves, a reminder of why we started the subproof. It is not formally part of the proof, and you can leave it out if you find it distracting.

To see how the rule works, suppose we want to prove a law of double negation $A \therefore \neg\neg A$

1	A	Want: $\neg\neg A$
2	$\neg A$	for reductio
3	A	R 1
4	$\neg A$	R 2
5	$\neg\neg A$	$\neg$I 2–4

The $\neg$E rule will work in much the same way. If we assume $\neg\mathcal{A}$ and show that it leads to a contradiction, we have effectively proven $\mathcal{A}$. So the rule looks like this:

m	$\neg\mathcal{A}$	for reductio		m	$\neg\mathcal{A}$	for reductio
n	$\mathcal{B}$			n	$\neg\mathcal{B}$	
$n+1$	$\neg\mathcal{B}$			$n+1$	$\mathcal{B}$	
$n+2$	$\mathcal{A}$	$\neg$E m–$n+1$		$n+2$	$\mathcal{A}$	$\neg$E m–$n+1$

Here is a simple example of negation elimination at work. We can show $L \leftrightarrow \neg O, L \vee \neg O \therefore L$ by assuming $\neg L$, deriving a contradiction, and then using $\neg$E.

1	$L \leftrightarrow \neg O$	
2	$L \vee \neg O$	Want: L
3	$\neg L$	for reductio
4	$\neg O$	$\vee$E 2, 3
5	L	$\leftrightarrow$E 1, 4
6	$\neg L$	R 3
7	L	$\neg$E 3–6

With the addition of $\neg$E and $\neg$I, our system of natural deduction is complete. We can now prove that any valid argument is actually valid. This is really where the fun begins.

One important bit of strategy. Sometimes, you will launch a subproof right away by assuming the negation of the conclusion to the whole argument. Other times, you will use a subproof to get

a piece of the conclusion you want, or some stepping stone to the conclusion you want. Here's a simple example. Suppose you were asked to show that this argument is valid:
$\neg(A \lor B) \therefore \neg A \land \neg B$. (The argument, by the way, is part of DeMorgan's Laws, some very useful equivalences which we will see more of later on.)

You need to set up the proof like this.

1 | $\neg(A \lor B)$ Want $\neg A \land \neg B$

Since you are trying to show $\neg A \land \neg B$, you could open a subproof with $\neg(\neg A \land \neg B)$ and try to derive a contradiction, but there is an easier way to do things. Since you are tying to prove a conjunction, you can set out to prove each conjunct separately. Each conjunct, then, would get its own reductio. Let's start by assuming A in order to show $\neg A$

1	$\neg(A \lor B)$	Want $\neg A \land \neg B$
2	A	for reductio
3	$A \lor B$	$\lor$I 2
4	$\neg(A \lor B)$	R 1
5	$\neg A$	$\neg$I 2–4

We can then finish the proof by showing $\neg B$ and putting it together with $\neg A$ and conjunction introduction.

1.5   Tautologies and Equivalences

So far all we've looked at is whether conclusions follow validly from sets of premises. However, as we saw in the chapter on truth tables, there are other logical properties we want to investigate: whether a statement is a tautology, a contradiction or a contingent statement, whether two statements are equivalent, and whether sets of sentences are consistent. In this section, we will look at using derivations to test for two properties which will be important in later sections, logical equivalence and being a tautology.

We can say that two statements are SYNTACTICALLY LOGICALLY EQUIVALENT IN SL if you can derive each of them from the other. We can symbolize this the same way we symbolized semantic equivalence. When we introduced the double turnstile, we said we would write the symbol facing both directions to indicate that two sentences were semantically equivalent, like this:
$A \land B =\!\models B \land A$. We can do the same thing with the single turnstile for syntactic equivalence, like this: $A \land B \dashv\vdash B \land A$.

For an example of how we can show two sentences to be syntactically equivalent, consider the sentences $P \to (Q \to R)$ and $(P \to Q) \to (P \to R)$. To prove these logically equivalent using

derivations, we simply use derivations to prove the equivalence one way, from P →(Q →R) to (P →Q) →(P →R). And then we prove it going the other way, from (P →Q) →(P →R) to P →(Q →R). We set up the proof going left to right like this:

$$1 \quad \boxed{P \to (Q \to R)} \qquad \text{Want: (P →Q) →(P →R)}$$

Since our want line is a conditional, we can set this up as a conditional proof. Once we set up the conditional proof, we also have a conditional in next want line, which means that we can put a conditional proof inside a conditional proof, like this.

1	$P \to (Q \to R)$	Want: (P →Q) →(P →R)
2	$\quad P \to Q$	Want: P →R
3	$\quad\quad P$	Want: R

The completed proof for the equivalence going in one direction will look like this.

1	$P \to (Q \to R)$	Want: (P →Q) →(P →R)
2	$\quad P \to Q$	Want: P →R
3	$\quad\quad P$	Want: R
4	$\quad\quad Q \to R$	→E 1, 3
5	$\quad\quad Q$	→E 2, 3
6	$\quad\quad R$	→E 4, 5
7	$\quad P \to R$	→I 3–6
8	$(P \to Q) \to (P \to R)$	→I 2–7

This shows that $P \to (Q \to R) \vdash (P \to Q) \to (P \to R)$. In order to show $P \to (Q \to R) \dashv\vdash (P \to Q) \to (P \to R)$, we need to prove the equivalence going the other direction. That proof will look like this:

```
1  | (P → Q) → (P → R)        Want: P →(Q →R)
2  |  | P                     Want: Q →R
3  |  |  | Q                  Want: R
4  |  |  |  | P               Want: Q
5  |  |  |  | Q               R 3
6  |  |  | P → Q              →I 4–5
7  |  |  | P → R              →E 1, 6
8  |  |  | R                  →E 2, 7
9  |  | Q → R                 →I 3–8
10 | P → (Q → R)             →I 2–9
```

You might think it is strange that we assume P twice in this proof, but that is the way we have to do it. When we assume P on line 2, our goal is to prove $P \to Q \to R$. Before we can finish that proof, we also need to know that $P \to Q$. This requires a different subproof.

These two proofs show that $P \to (Q \to R)$ and $(P \to Q) \to (P \to R)$ are equivalent, so we can write $P \to (Q \to R) \dashv\vdash (P \to Q) \to (P \to R)$.

We can also prove that a sentence is a tautology using a derivation. A tautology is something that must be true as a matter of logic. If we want to put this in syntactic terms, we would say that SYNTACTIC TAUTOLOGY IN SL is a statement that can be derived without any premises, because its truth doesn't depend on anything else. Now that we have all of our rules for starting and ending subproofs, we can actually do this. Rather than listing any premises, we simply start a subproof at the beginning of the derivation. The rest of the proof can work only using premises assumed for the purposes of subproofs. By the end of the proof, you have discharged all these assumptions, and are left knowing a tautological statement without relying on any leftover premises. Consider this proof of the law of noncontradiction: $\neg(G \land \neg G)$.

```
1  |  | G ∧ ¬G         for reductio
2  |  | G              ∧E 1
3  |  | ¬G             ∧E 1
4  | ¬(G ∧ ¬G)         ¬I 1–3
```

This statement simply says that any sentence G cannot be both true and not true at the same time. We prove it by imagining what would happen if G were actually both true and not true, and then pointing out that we already have our contradiction.

In the previous chapter, we expressed the fact that something could be proven a tautology using truth tables by writing the double turnstile in front of it. The law of noncontradiction above

could have been proven using truth tables, so we could write: $\models \neg(G \wedge \neg G)$ In this chapter, we will use the single turnstile the same way, to indicate that a sentence can be proven to be a tautology using a derivation. Thus the above proof entitles us to write $\vdash \neg(G \wedge \neg G)$.

1.6   Derived Rules

Now that we have our five rules for introduction and our five rules for elimination, plus the rule of reiteration, our system is complete. If an argument is valid, and you can symbolize that argument in SL, you can *prove* that the argument is valid using a derivation. (We will say a bit more about this in section 1.7.) Now that our system is complete, we can really begin to play around with it and explore the exciting logical world it creates.

There's an exciting logical world created by these eleven rules? Yes, yes there is. You can begin to see this by noticing that there are a lot of other interesting rules that we could have used for our introduction and elimination rules, but didn't. In many textbooks, the system of natural deduction has a disjunction elimination rule that works like this:

$$
\begin{array}{r|l}
m & \mathcal{A} \vee \mathcal{B} \\[4pt]
n & \mathcal{A} \rightarrow \mathcal{C} \\[4pt]
o & \mathcal{B} \rightarrow \mathcal{C} \\[4pt]
 & \mathcal{C} \qquad\qquad \vee * \; m,\, n,\, o
\end{array}
$$

You might think our system is incomplete because it lacks this alternative rule of disjunction elimination. Yet this is not the case. If you can do a proof with this rule, you can do a proof with the basic rules of the natural deduction system. You actually proved this rule in problem **??** of part **??** in the exercises for section 1.4. Furthermore, once you have a proof of this rule, you can use it inside other proofs whenever you think you would need a rule like $\vee *$. Simply use the proof you gave in the last homework as a sort of recipe for generating a new series of steps to get you to a line saying $\mathcal{C} \vee \mathcal{D}$

But adding lines to a proof using this recipe all the time would be a pain in the neck. What's worse, there are dozens of interesting possible rules out there, which we could have used for our introduction and elimination rules, and which we now find ourselves replacing with recipes like the one above.

Fortunately our basic set of introduction and elimination rules, plus reiteration, was meant to be expanded on. That's part of the game we are playing here. The first system of deduction created in the Western tradition was the system of geometry created by Euclid (c 300 BCE). Euclid's *Elements* began with 10 basic laws, along with definitions of terms like "point," "line," and "plane." He then went on to prove hundreds of different theorems about geometry, and each time he proved a theorem he could use that theorem to help him prove later theorems.

We can do the same thing in our system of natural deduction. What we need is a rule that will allow us to make up new rules. The new rules we add to the system will be called DERIVED RULES. Our ten rules for adding and eliminating connectives are then the AXIOMS of SL. Now here is our rule for adding rules.

> **Rule of Derived Theorem Introduction:** Given a derivation in SL of some argument $A_1 \ldots A_n \vdash B$, create the rule $\mathcal{A}_1 \ldots \mathcal{A}_n \vdash \mathcal{B}$ and assign a name to it of the form "T_n", to be read "theorem n." Now given a derivation of some theorem T_m, where $n < m$, if $\mathcal{A}_1 \ldots \mathcal{A}_n$ occur as earlier lines $x_1 \ldots x_n$ in a proof, one may infer $\mathcal{B}$, and justify it "$T_n, x_1 \ldots x_n$", so long as none of lines $x_1 \ldots x_n$ are in a closed subproof.

Let's make our rule $\lor*$ above our first theorem. The proof of T_1 is derived simply from the recipe above.

T_1 **(Constructive Dilemma, CD):** $\{\mathcal{A} \lor \mathcal{B}, \mathcal{A} \to \mathcal{C}, \mathcal{B} \to \mathcal{C}\} \vdash \mathcal{C}$

Proof:

1	$A \lor B$	
2	$A \to C$	
3	$B \to C$	want: C
4	$\neg C$	for reductio
5	$\neg A$	for reductio
6	B	$\lor$E 2, 5
7	C	$\to$I 3, 6
8	$\neg C$	R 4
9	A	$\neg$I 5–8
10	C	$\to$E 2, 10
11	$\neg C$	R 4
12	C	$\neg$I 4–12

Informally, we will refer to T_1 as "Constructive Dilemma" or by the abbreviation "CD." Most theorems will have names and easy abbreviations like this. We will generally use the abbreviations to refer to the proofs when we use them in derivations, because they are easier to remember.

Several other important theorems have already appeared as examples or in homework problems. We'll talk about most of them in the next section, when we discuss rules of replacement. In the meantime, there is one important one we need to introduce now

T_2 **(Modus Tollens, MT):** $\{\mathcal{A} \to \mathcal{B}, \neg \mathcal{B}\} \vdash \neg \mathcal{A}$

Proof: See page **??**

Now that we have some theorems, let's close by looking at how they can be used in a proof.

T$_3$ (Destructive Dilemma, DD): $\{\mathcal{A} \to \mathcal{B}, \mathcal{A} \to \mathcal{C}, \neg\mathcal{B} \vee \neg\mathcal{C}\} \vdash \neg\mathcal{A}$

1	$A \to B$	
2	$A \to C$	
3	$\neg B \vee \neg C$	Want: ¬A
4	A	for reductio
5	B	→E 1, 4
6	$\neg B$	For reductio
7	B	R 5
8	$\neg B$	R 6
9	$\neg\neg B$	¬I 6–8
10	$\neg C$	∨E 3, 9
11	A	R 4
12	$\neg A$	MT 2, 10
13	$\neg A$	¬I 4–12

1.7   Rules of Replacement

Very often in a derivation, you have probably been tempted to apply a rule to a part of a line. For instance, if you knew $F \to (G \wedge H)$ and wanted $F \to G$, you would be tempted to apply ∧E to just the $G \wedge H$ part of $F \to (G \wedge H)$. But, of course you aren't allowed to do that. We will now introduce some new derived rules where you can do that. These are called RULES OF REPLACEMENT, because they can be used to replace part of a sentence with a logically equivalent expression. What makes the rules of replacement different from other derived rules is that they draw on only one previous line and are symmetrical, so that you can reverse premise and conclusion and still have a valid argument. Some of the most simple examples are Theorems $8 - 10$, the rules of commutativity for $\wedge$, $\vee$, and $\leftrightarrow$.

T$_4$ (Commutativity of $\wedge$, Comm$\wedge$): $(\mathcal{A} \wedge \mathcal{B}) \dashv\vdash (\mathcal{B} \wedge \mathcal{A})$
T$_5$ (Commutativity of $\vee$, Comm$\vee$): $(\mathcal{A} \vee \mathcal{B}) \dashv\vdash (\mathcal{B} \vee \mathcal{A})$
T$_6$ (Commutativity of $\leftrightarrow$, Comm$\leftrightarrow$): $(\mathcal{A} \leftrightarrow \mathcal{B}) \dashv\vdash (\mathcal{B} \leftrightarrow \mathcal{A})$

You will be asked to prove these in the homework. In the meantime, let's see an example of

how they work in a proof. Suppose you wanted to prove $(M \vee P) \to (P \wedge M)$, $\therefore$
$(P \vee M) \to (M \wedge P)$ You could do it using only the basic rules, but it will be long and
inconvenient. With the Comm rules, we can provide a proof easily:

1	$(M \vee P) \to (P \wedge M)$	
2	$(P \vee M) \to (P \wedge M)$	Comm$\wedge$ 1
3	$(P \vee M) \to (M \wedge P)$	Comm$\vee$ 2

Formally, we can put our rule for deploying rules of replacement like this

Inserting rules of replacement: Given a theorem T of the form $\mathcal{A} \dashv\vdash \mathcal{B}$ and a line
in a derivation $\mathcal{C}$ which contains in it a sentence $\mathcal{D}$, where $\mathcal{D}$ is a substitution instance
of either $\mathcal{A}$ or $\mathcal{B}$, replace $\mathcal{D}$ with the equivalent substitution instance of the other side of
theorem T.

Here are some other important theorems that can act as rules of replacement. Some are
theorems we have already proved, while you will be asked to prove others in the homework.

T$_7$ (Double Negation, DN): $\mathcal{A} \dashv\vdash \neg\neg\mathcal{A}$

Proof: See pages 16 and **??**.

T$_8$: $\neg(\mathcal{A} \vee \mathcal{B}) \dashv\vdash \neg\mathcal{A} \wedge \neg\mathcal{B}$

Proof: See page **??**

T$_9$: $\neg(\mathcal{A} \wedge \mathcal{B}) \dashv\vdash \neg\mathcal{A} \vee \neg\mathcal{B}$

Proof: See pages **??** and **??**.

T_{12} and T_{13} are collectively known as DEMORGAN'S LAWS, and we will use the abbreviation
DeM to refer to either of them in proofs.

T$_{10}$: $(\mathcal{A} \to \mathcal{B}) \dashv\vdash (\neg\mathcal{A} \vee \mathcal{B})$

T$_{11}$: $(\mathcal{A} \vee \mathcal{B}) \dashv\vdash (\neg\mathcal{A} \to \mathcal{B})$

T_{14} and T_{15} are collectively known as the rule of Material Conditional (MC). You will prove
them in the homework.

T$_{12}$ (Biconditional Exportation, ex): $\mathcal{A} \leftrightarrow \mathcal{B} \dashv\vdash (\mathcal{A} \to \mathcal{B}) \wedge (\mathcal{B} \to \mathcal{A})$

Proof: See the homework.

T$_{13}$ (Contraposition, trans): $\mathcal{A} \to \mathcal{B} \dashv\vdash \neg\mathcal{B} \to \neg\mathcal{A}$

Proof: See the homework.

To see how much these theorems can help us, consider this argument:

$$\neg(P \rightarrow Q) \vdash P \wedge \neg Q$$

As always, we could prove this argument using only the basic rules. With rules of replacement, though, the proof is much simpler:

1	$\neg(P \rightarrow Q)$	
2	$\neg(\neg P \vee Q)$	MC 1
3	$\neg\neg P \wedge \neg Q$	DeM 2
4	$P \wedge \neg Q$	DN 3

1.8   Proof Strategy

There is no simple recipe for proofs, and there is no substitute for practice. Here, though, are some rules of thumb and strategies to keep in mind.

Work backwards from what you want. The ultimate goal is to derive the conclusion. Look at the conclusion and ask what the introduction rule is for its main logical operator. This gives you an idea of what should happen *just before* the last line of the proof. Then you can treat this line as if it were your goal. Ask what you could do to derive this new goal. For example: If your conclusion is a conditional $\mathcal{A} \rightarrow \mathcal{B}$, plan to use the $\rightarrow$I rule. This requires starting a subproof in which you assume $\mathcal{A}$. In the subproof, you want to derive $\mathcal{B}$. Similarly, if your conclusion is a biconditional, $\mathcal{A} \leftrightarrow \mathcal{B}$, plan on using $\leftrightarrow$I and be prepared to launch two subproofs. If you are trying to prove a single sentence letter or a negated single sentence letter, you might plan on using indirect proof.

Work forwards from what you have. When you are starting a proof, look at the premises; later, look at the sentences that you have derived so far. Think about the elimination rules for the main operators of these sentences. These will tell you what your options are. For example: If you have $A \wedge B$ use $\wedge$E to get A and B separately. If you have $A \vee B$ see if you can find the negation of either A or B and use $\vee$E.

Repeat as necessary. Once you have decided how you might be able to get to the conclusion, ask what you might be able to do with the premises. Then consider the target sentences again and ask how you might reach them. Remember, a long proof is formally just a number of short proofs linked together, so you can fill the gap by alternately working back from the conclusion and forward from the premises.

Change what you are looking at. Replacement rules can often make your life easier. If a proof seems impossible, try out some different substitutions.For example: It is often difficult to prove a disjunction using the basic rules. If you want to show $\mathcal{A} \vee \mathcal{B}$, it is often easier to show $\neg\mathcal{A} \rightarrow \mathcal{B}$ and use the MC rule. Some replacement rules should become second nature. If you see a negated disjunction, for instance, you should immediately think of DeMorgan's rule.

When all else fails, try indirect proof. If you cannot find a way to show something directly, try assuming its negation. Remember that most proofs can be done either indirectly or directly. One way might be easier—or perhaps one sparks your imagination more than the other—but either one is formally legitimate.

Take a break If you are completely stuck, put down your pen and paper, get up from your computer, and do something completely different for a while. Walk the dog. Do the dishes. Take a shower. I find it especially helpful to do something physically active. Doing other desk work or watching TV doesn't have the same effect. When you come back to the problem, everything will seem clearer. Of course, if you are in a testing situation, taking a break to walk around might not be advisible. Instead, switch to another problem.

A lot of times, when you are stuck, your mind keeps trying the same solution again and again, even though you know it won't work. "If I only knew $Q \rightarrow R$," you say to yourself, "it would all work. Why can't I derive $Q \rightarrow R$!" If you go away from a problem and then come back, you might not be as focused on That One Thing that you were sure you needed, and you can find a different approach.

Key Terms

Completeness

Soundness

Syntactically consistent in SL

Syntactically contingent in SL

Syntactically inconsistent in SL

Syntactically logically equivalent in SL

Syntactically valid in SL

Syntactic contradiction in SL

Syntactic tautology in SL

Bibliography