Chapter 2

INTRODUCTION TO DEMONSTRATIONS

2.1 Modus Ponens; A First Look at Demonstrations

We are now ready to begin to use the propositional calculus to study mathematical arguments. What we propose to do is to look at a few arguments to gain some ideas about what we might incorporate into the propositional calculus. We will then develop the propositional calculus further. Let us begin by explaining our overall plan.

Our objective is to analyze mathematical arguments, seeking to understand why we find some arguments convincing and others not, or why we accept some arguments as sound and reject other arguments as unsound. To begin with, we contend that there are tacit assumptions (that is, assumptions that you don't bother to mention because they're "obvious") in most arguments; indeed, most human communication involves much that is tacit. Let us illustrate.

Suppose that someone is told, "See if it is raining." That simple request may trigger a very complex response. Suppose the person is seated, reading a book. He puts down his book, gets up from the chair, walks to the window, opens the drapes, and looks out. He knows exactly what to do without being told. Indeed if he were told, "Put down your book, get up from the chair, walk to the window, open the drapes, look out of the window, and see if it is raining", he would think it very strange to be told all that obvious stuff.

Of course we have all had the experience of being asked to do something and not being able to do it because the instructions were not sufficient to enables us to understand what to do. Herein lies the artistry of mathematical argument. A good argument should say enough so that it clearly understood but not so much that we are bored by unnecessary details. But how much is enough? That depends on the audience. Someone who knows a great deal needs less information than someone who knows very little.

Suppose that you are trying to convince someone of something and they cannot follow your argument. What do you do? You give a more detailed argument. What we propose to do in analyzing mathematical arguments is to insert sufficient details that the steps of the argument can be followed in a purely mechanical way; that is, the argument could be checked by a suitably designed machine. Why do we want to do this? Certainly not because we think we should all strive to imitate machines, but rather because it presents a very thorough analysis of an argument and provides us with a very high standard to set for mathematical arguments. We propose the following as our first approximation to the idea of proof.

We will call an argument a <u>proof</u> provided we can, with reasonable ease, insert sufficient details that the argument can be checked by a suitably designed machine. We must of course reach some agreement about what capabilities we should build into the design of our imaginary machine so that it checks what we want it to check. To help us reach a decision on this question we will examine several arguments and pick out what seem to be the important aspects of a convincing argument. Let us begin by looking at the argument on page 13 of Book 0, Chapter 1.

The author argues that from the fact that $96 = 3 \cdot 29 + 9$ we can tell "everything about the position of 96 on our spiral that we need to know. In the first place, we see that 96 is on the fourth winding since the fourth winding begins at $3 \cdot 29$." We hope you agree with us that the argument is quite convincing. Nevertheless it involves an unstated premise: if a number lies between $3 \cdot 29$ and $3 \cdot 29 + 28$ inclusive, then that number lies on the fourth winding. In particular:

> If 96 lies between $3 \cdot 29$ and $3 \cdot 29 + 28$ inclusive,
>
> then 96 lies on the fourth winding.
>
> But 96 does lie between $3 \cdot 29$ and $3 \cdot 29 + 28$ inclusive
>
> (since $3 \cdot 29 = 87$ and $3 \cdot 29 + 28 = 115$).
>
> Therefore 96 lies on the fourth winding.

The author continues: "3 • 29 (or 87) is the first number of a winding, so it corresponds to 0 on the inner winding. Therefore 3 • 29 + 9 (or 96) must correspond to 9 on the inner winding." Again there is an unstated premise: If a number is the 10th number of a winding, then that number corresponds to 9 on the inner winding. In particular:

> If 96 is the 10th number of the fourth winding,
>
> then 96 corresponds to 9 on the inner winding.
>
> But 96 is the 10th number of the fourth winding.
>
> Therefore 96 corresponds to 9 on the inner winding.

Notice the similarity in these two arguments. They each follow the same general pattern:

$$P \Rightarrow Q$$

$$P$$

Therefore $\quad Q$.

This argument pattern is an example of what is known as a <u>rule of inference</u>. From given hypotheses[*] one draws a conclusion. In this particular case the hypotheses are "$P \Rightarrow Q$" and "P" and the conclusion is "Q". This particular rule of inference is called <u>Modus Ponens</u>[◇]. Although the example before us involves only atomic formulas, "P" and "Q", that is not required. More generally the rule of Modus Ponens is the following.

Modus Ponens: If $\mathcal{S}$ and $\mathcal{T}$ are wff's, then from

$$\mathcal{S} \Rightarrow \mathcal{T} \quad \text{and} \quad \mathcal{S}$$

we may conclude $\mathcal{T}$.

Of the two hypotheses, the conditional, $\mathcal{S} \Rightarrow \mathcal{T}$, is called the <u>major premise</u> and $\mathcal{S}$ is called the <u>minor premise</u>.

---

[*] At some time or another you have probably been involved in a discussion in which you have been told, "Let's assume for the sake of argument that..." In such a situation you are not being asked actually to <u>believe</u> that something is true, it's more a case of being asked to make a temporary supposition so that the argument can get started. It is in this sense that we make assumptions in mathematical arguments. Throughout this book we will usually refer to such an assumption by its technical name: <u>hypothesis</u> (plural: hypotheses).

[◇] In classical logic the full name for the Modus Ponens rule of inference is "Modus Ponendo Ponens", which may literally be translated as "The rule (modus) that affirms (ponens) by affirming (ponendo)"; the meaning being that, given the conditional "$P \Rightarrow Q$", if you affirm (i.e., accept) the antecedent, "P", then you may also affirm the consequent, "Q".

We will use Modus Ponens extensively in our analysis of mathematical arguments. For our purposes it is a good rule. For one thing, from a rule of inference we want confidence in the conclusion drawn; that is, confidence that, if the hypotheses are true, then so is the conclusion. Clearly Modus Ponens is a good rule on this count because from truth table ($\mathcal{I}$) we know that if $\mathcal{S} \Rightarrow \mathcal{T}$ and $\mathcal{S}$ both have truth value t, then $\mathcal{T}$ must necessarily also have truth value t.

Modus Ponens is a good rule on another count. It is a simple rule to check mechanically. Let us imagine a machine that can read and print the symbols of the propositional calculus. Imagine that the machine is capable of checking a formula to see if it is well formed and of comparing formulas to see if they are the same or different. In order for our machine to do something for us we must give it information on which to work and instructions as to what we wish it to do. If, for example, we want the machine to perform a Modus Ponens operation, we must first feed into the machine the formulas on which it is to operate. We will refer to these formulas as <u>hypotheses</u>. We put the hypotheses into the machine by means of an input storage order. Such orders look like this

Hyp, $P \Rightarrow Q$, P.

When the machine receives this message, it reads "Hyp" and from that it learns that what follows is a sequence of formulas to be stored away. The period at the end tells the machine that that is the end of the sequence. However, before the machine stores those formulas it first checks them to make sure they are well formed. This is simply a check for any mistakes that might have been made in typing the input storage order. If the machine finds that an input formula is not well formed a red light flashes and the machine prints out the faulty formula followed by the symbols

~wff.

To recall a formula from storage we must give the machine a print order. Now the machine remembers what it prints and the order in which the printing is done. For example, if we give it the print order

Pr, Hyp 2.

the machine reads this as an instruction to print the second hypothesis of the input storage order, and so it prints

1) P.

Notice that although we asked the machine to print the second hypothesis it labeled its response " 1) ". This is because "P" is the first formula that the machine has printed since it began working on this program. Indeed, for as long as the machine is under orders from this program it will remember "P" as " For 1 " (which is machine language for "formula (1)").
If we follow the first print order by

Pr, Hyp 1.

the machine will print

2) $P \Rightarrow Q$

which it will remember as " For 2 ". If we wish the machine to apply Modus Ponens to For 1 and For 2, we must give it the order

(*) MP, For 2, For 1.

On receiving this order the machine examines the first formula mentioned in (*), namely For 2, to make sure that it is a conditional. If it is, the machine then compares its antecedent with the second formula mentioned, namely For 1. If these are the same, the machine prints out the consequent of the conditional, For 2. Thus, for the example at hand, the machine responds to order (*) by printing

3) Q.

If however the machine had been given input formulas so that For 2 was not a conditional or its antecedent was not the same as For 1, then the machine would respond to order (*) by flashing its red light and printing out

For incom

(which is machine language for "the formulas are incompatible for Modus Ponens").

If you reflect on what the machine does you will see that it is very important when giving a MP order to mention the major premise first and the minor premise second. If you switch them you will get a red light. It would not be too difficult to modify our machine so that it would accept the major and minor premises in either order, but such considerations need not concern us now.

At this point let us look at another example of a mathematical argument. We are still not completely ready to analyze such arguments. Nevertheless from this example we will be able to indicate our plan of attack and illustrate the next step we must take in developing the propositional calculus.

On page 110 of Book 0, Chapter 1, the author proves that an operational system cannot have two neutral elements. The essence of the argument is that if n_1 and n_2 are neutral elements, then $n_1 * n_2 = n_2$ and $n_1 * n_2 = n_1$ and hence $n_1 = n_2$. We might begin to analyze this argument in the following way:

Step		Formula
(1)	If n_1 is a neutral element then $n_1 * n_2 = n_2$	$P \Rightarrow S$
(2)	But n_1 is a neutral element.	P
(3)	Therefore $n_1 * n_2 = n_2$	S
(4)	If n_2 is a neutral element then $n_1 * n_2 = n_1$	$Q \Rightarrow T$
(5)	But n_2 is a neutral element.	Q
(6)	Therefore $n_1 * n_2 = n_1$.	T
(7)	If $n_1 * n_2 = n_2$ and $n_1 * n_2 = n_1$, then $n_1 = n_2$	$[S \wedge T] \Rightarrow R$
(8)	But $n_1 * n_2 = n_2$ and $n_1 * n_2 = n_1$.	$S \wedge T$
(9)	Therefore $n_1 = n_2$.	R

There are many interesting issues to discuss here but for the moment we only wish to consider the sequence of wffs in the right hand column. This sequence of formulas is an example of what we call a <u>machine demonstration</u> or simply a <u>demonstration</u>. It is a demonstration of "R". When viewed in isolation from the meaning of the sentences to which they correspond, each of the formulas in this demonstration is either a hypothesis or it follows by Modus Ponens from earlier steps: Formulas (1) and (2)[*] are hypotheses, but

---

* We give each formula the same number as that of the step to which it corresponds. For example, "$Q \Rightarrow T$" corresponds to step (4) and so will be referred to as "formula (4)".

formula (3) follows from formulas (1) and (2) by Modus Ponens. Formulas (4) and (5) are hypotheses, but formula (6) follows from formulas (4) and (5) by Modus Ponens. Formulas (7) and (8) are hypotheses, but formula (9) follows from formulas (7) and (8) by Modus Ponens. Surely anyone who accepts those six hypotheses as true will be convinced of the truth of the conclusion of the demonstration, formula (9).

You would be perfectly justified at this point to object, "Yes, but you're asking us to accept quite a lot. Of the nine formulas in the demonstration six are assumptions. It would be much more convincing if you were able to provide a demonstration that didn't require us to assume quite so much." This is a general problem in the construction of mathematical arguments, and indeed the question of reducing the number of hypotheses in an argument is an issue that we will wish to explore further at a later time and about which we have one observation to make now. We can eliminate the hypothesis stated as formula (8) if we permit ourselves the use of the tautology

(8′) $\quad S \Rightarrow [T \Rightarrow [S \wedge T]].$

From formulas (8′) and (3) we have, by Modus Ponens,

(9′) $\quad T \Rightarrow [S \wedge T].$

Then, using Modus Ponens on formulas (9′) and (6), we have

(10′) $\quad S \wedge T.$

Finally from formulas (10′) and (7) we have, again by Modus Ponens,

(11′) $\quad R.$

We have lengthened the demonstration and eliminated a hypothesis. It is a fair trade. Thus the hypothesis of formula (8) can be avoided if we accept the tautology of formula (8′). Since tautologies are unconditionally true no one will object to their use. With that agreement we are ready to say exactly what we are going to mean by a "demonstration".

Let $\mathcal{S}$, $\mathcal{H}_1, \ldots, \mathcal{H}_n$ be well formed formulas of the propositional calculus. By a demonstration of $\mathcal{S}$ from the hypotheses $\mathcal{H}_1, \ldots, \mathcal{H}_n$, we mean a sequence of wffs $\mathcal{S}_1, \ldots, \mathcal{S}_k$ such that $\mathcal{S}_k$ is $\mathcal{S}$ and for each $\mathcal{S}_i$ in the sequence one of the following is the case:

1) $\mathcal{S}_i$ is one of the hypotheses;

2) $\mathcal{S}_i$ is a tautology;

3) $\mathcal{S}_i$ follows from two earlier formulas of the sequence by Modus Ponens.

If there is a demonstration of $\mathcal{S}$ from hypotheses $\mathcal{H}_1, \ldots, \mathcal{H}_n$, then we say that $\mathcal{S}$ is a logical consequence of $\mathcal{H}_1, \ldots, \mathcal{H}_n$, or that $\mathcal{S}$ is deducible from $\mathcal{H}_1, \ldots, \mathcal{H}_n$, or simply that $\mathcal{S}$ follows from $\mathcal{H}_1, \ldots, \mathcal{H}_n$. The assertion that such a demonstration exists may be expressed symbolically thus:

$$\mathcal{H}_1, \ldots, \mathcal{H}_n \vdash \mathcal{S}$$

which can be read in any of the ways just described or simply as "$\mathcal{H}_1, \ldots, \mathcal{H}_n$ yields $\mathcal{S}$". The symbol "$\vdash$" is called a turnstile. As you will see in the examples to follow some formulas follow from no hypotheses at all. This may be expressed symbolically thus:

$$\vdash \mathcal{S}$$

which we propose to read as "$\mathcal{S}$ is deducible from no hypotheses" or simply "$\mathcal{S}$ is deducible".

Suppose that $\mathcal{S}$ is deducible from $\mathcal{H}_1, \ldots, \mathcal{H}_n$, i.e., $\mathcal{H}_1, \ldots, \mathcal{H}_n \vdash \mathcal{S}$. What practical significance does this information have? To answer this question let us reflect on what we know once we know that a demonstration of $\mathcal{S}$ from the hypotheses $\mathcal{H}_1, \ldots, \mathcal{H}_n$ exists. Such a demonstration consists of a sequence of formulas which are of three kinds: hypotheses, tautologies, and the results of applying Modus Ponens. Now tautologies are unconditionally true. So if, more than just having been assumed "for the sake of argument", the hypotheses are formulas that actually have truth value t, then examination of truth table ($\mathcal{I}$) reveals that those formulas which result from the application of Modus Ponens must also have truth value t.

Hence, in such a case every formula in the demonstration—and in particular $\mathcal{S}$—has truth value t. This result may be stated as follows: A false conclusion cannot follow logically from true hypotheses. So, given an interesting set of hypotheses, it seems that it might also be interesting to know all the well formed formulas that are deducible from those hypotheses. But now we are getting ahead of ourselves. Let us return to our definition of a demonstration, examine several examples, do lots of exercises, and be sure we understand it very, very well.

Example 1: As our first example we will reexamine a result that we considered earlier in this section.

$P \Rightarrow Q$

P

Q

is a demonstration of "Q" from the hypotheses " $P \Rightarrow Q$ " and "P". The existence of this demonstration confirms that $P \Rightarrow Q,\ P \vdash Q$ ⚹.

Earlier in this section we referred to a demonstration as a "machine demonstration". This was simply because a machine could check a demonstration to make sure it obeys the rules we have introduced so far. All we need do is give the machine all the information it needs. For a machine check of the demonstration of Example 1 we begin with an input storage order:

Hyp, $P \Rightarrow Q$, P.

We follow this with a sequence of print orders that we call a program for the demonstration:

Pr, Hyp 1

Pr, Hyp 2

MP, For 1, For 2.

The machine carries out its instructions and prints

⚹ We will have something to say about this notation in Section 2.9.

1) $P \Rightarrow Q$

2) P

3) Q.

Example 2: Show that $P \wedge Q \vdash P$.

	Demonstration	Program Outline
1)	$P \wedge Q$	Hyp
2)	$[P \wedge Q] \Rightarrow P$	Taut
3)	P	MP, For 2, For 1

Let us explain the column headed "Program Outline". Our intention is to provide sufficient information in this column that it becomes a trivial matter to write a program that will cause our machine to check the demonstration; the check, of course, consists in feeding the machine the program and seeing if our demonstration is what is printed. The notation "Hyp" on the first line indicates that " $P \wedge Q$ " is a hypothesis and therefore needs to be stored in the machine by means of an input storage order:

Hyp, $P \wedge Q$.

It also reminds us that the first step of our program should be

Pr, Hyp 1.

The notation "Taut" on the second line indicates that " $[P \wedge Q] \Rightarrow P$ " is a tautology. Now our machine needs to be told that this formula is a tautology that we are going to use in this demonstration, and this is done by means of a second type of input order:

Taut, $[P \wedge Q] \Rightarrow P$.

(In passing, let us remark that, unfortunately, our machine has no way of recognizing tautologies; it blindly accepts the word of the programmer that whatever is input as a tautology is indeed a tautology. Hence it is the responsibility of the programmer to verify that a given formula is a tautology before presenting it to our poor, trusting machine as such.) In the same way that the machine remembers hypotheses in the order in which they are introduced by hypothesis input storage orders, it also remembers tautologies in the order in which they are introduced by tautology input storage orders. Hence, when we want (as we do in the second step of our program) to order the machine to print this tautology, we tell it to print the first (and in this case, the only) tautology that was stored. This is done by means of the following print order:

Pr, Taut 1.

Bearing in mind these words of explanation, it is then very easy to write out this program:

Pr, Hyp 1

Pr, Taut 1

MP, For 2, For 1.

Example 3: Show that $P, Q \vdash P \wedge Q$.

	Demonstration	Program Outline
1)	P	Hyp
2)	Q	Hyp
3)	$P \Rightarrow [Q \Rightarrow [P \wedge Q]]$	Taut*
4)	$Q \Rightarrow [P \wedge Q]$	MP, For 3, For 1
5)	$P \wedge Q$	MP, For 4, For 2

Example 4: Show that $\vdash P \vee {\sim}P$.

	Demonstration	Program Outline
1)	$P \vee {\sim}P$	Taut

* Don't forget: it's up to you to check that this really is a tautology. We won't remind you again, so from here on make sure each time that we have written "Taut" in a program outline that we are not misleading the machine.

Example 5: Show that $P \Rightarrow Q,\ Q \Rightarrow R \vdash P \Rightarrow R$.

	Demonstration	Program Outline
1)	$P \Rightarrow Q$	Hyp
2)	$Q \Rightarrow R$	Hyp
3)	$[P \Rightarrow Q] \Rightarrow [[Q \Rightarrow R] \Rightarrow [P \Rightarrow R]]$	Taut
4)	$[Q \Rightarrow R] \Rightarrow [P \Rightarrow R]$	MP, For 3, For 1
5)	$P \Rightarrow R$	MP, For 4, For 2

Example 6: Show that $S \Rightarrow T,\ S \vdash \sim[\sim T]$.

	Demonstration	Program Outline
1)	$S \Rightarrow T$	Hyp
2)	S	Hyp
3)	T	MP, For 1, For 2
4)	$T \Rightarrow \sim[\sim T]$	Taut
5)	$\sim[\sim T]$	MP, For 4, For 3

Example 7: Show that $P \Rightarrow \sim R,\ Q \Rightarrow P,\ Q \vdash \sim R$.

	Demonstration	Program Outline
1)	$P \Rightarrow \sim R$	Hyp
2)	$Q \Rightarrow P$	Hyp
3)	Q	Hyp
4)	P	MP, For 2, For 3
5)	$\sim R$	MP, For 1, For 4

PROBLEM SET 6

In Problems 1 - 3, complete the program outline for the given demonstration.

1. Show that $P \vdash P \vee R$.

	Demonstration	Program Outline
1)	P	
2)	$P \Rightarrow [P \vee R]$	
3)	$P \vee R$	

2. Show that $[[P \wedge Q] \vee {}^{\sim}[P \wedge Q]] \Rightarrow R \vdash R$.

	Demonstration	Program Outline
1)	$[[P \wedge Q] \vee {}^{\sim}[P \wedge Q]] \Rightarrow R$	
2)	$[P \wedge Q] \vee {}^{\sim}[P \wedge Q]$	
3)	R	

3. Show that $P,\ P \Leftrightarrow Q \vdash Q$.

	Demonstration	Program Outline
1)	P	
2)[*]	$[P \Rightarrow Q] \wedge [Q \Rightarrow P]$	
3)	$[[P \Rightarrow Q] \wedge [Q \Rightarrow P]] \Rightarrow [P \Rightarrow Q]$	
4)	$P \Rightarrow Q$	
5)	Q	

In Problems 4 - 6, fill in the missing steps of the demonstration and complete the program outline.

4. Show that ${}^{\sim}[P \wedge {}^{\sim}P] \Rightarrow Q \vdash Q$.

	Demonstration	Program Outline
1)	${}^{\sim}[P \wedge {}^{\sim}P] \Rightarrow Q$	
2)		Taut
3)	Q	

---

[*] Recall that " $P \Leftrightarrow Q$ " is a shorthand for " $[P \Rightarrow Q] \wedge [Q \Rightarrow P]$ ". So, in a given situation, we are at liberty to write one or the other, whichever is the more convenient for the purpose at hand.

5. Show that $P, \sim Q \vdash \sim[P \Rightarrow Q]$.

Demonstration		Program Outline
1)		Hyp
2)		Hyp
3)	$P \Rightarrow [\sim Q \Rightarrow [P \wedge \sim Q]]$	
4)		MP, For 3, For 1
5)	$P \wedge \sim Q$	
6)	$[P \wedge \sim Q] \Rightarrow \sim[\sim P \vee Q]$	
7)		MP, For 6, For 5
8)		Taut
9)	$\sim[P \Rightarrow Q]$	

6. Show that $P, Q, R, [P \wedge Q] \Rightarrow S, [S \wedge R] \Rightarrow T \vdash T$.

Demonstration		Program Outline
1)	P	
2)	Q	
3)	$P \Rightarrow [Q \Rightarrow [P \wedge Q]]$	
4)		MP, For 3, For 1
5)		MP, For 4, For 2
6)		Hyp
7)	S	
8)	$S \Rightarrow [R \Rightarrow [S \wedge R]]$	
9)	$R \Rightarrow [S \wedge R]$	
10)		Hyp
11)	$S \wedge R$	
12)		Hyp
13)	T	

In Problems 7 - 10, give an appropriate demonstration and program outline.

7. Show that $P \wedge Q \vdash P$.

8. Show that $P \Rightarrow [Q \wedge R]$, $P \wedge Q \vdash R$.

9. Show that $P \Rightarrow Q \vdash \sim Q \Rightarrow \sim P$.

10. Show that $Q \vdash P \Rightarrow Q$.

- o O o -

Aftermath...

A drunkard had read so much about the evils of drinking,
he finally decided to give up reading.

- o O o -

2.2 Conjunctive Inference and Conjunctive Simplification

If you look back on what we have been doing you will see that in forming demonstrations we are doing all the work; the machine is merely checking what we have already done. Surely we should be able to put the machine to better use. This is indeed the case, for if we examine the demonstrations we have made thus far we note that there are certain steps that show up repeatedly from demonstration to demonstration. For example, in many demonstrations we need to infer $\mathcal{S} \wedge \mathcal{T}$ from $\mathcal{S}$ and from $\mathcal{T}$, where $\mathcal{S}$ and $\mathcal{T}$ are well formed formulas. To do this we introduce the tautology

$$\mathcal{S} \Rightarrow [\mathcal{T} \Rightarrow [\mathcal{S} \wedge \mathcal{T}]]$$

from which, by two applications of Modus Ponens, we conclude $\mathcal{S} \wedge \mathcal{T}$. We could modify the machine so that it would do all three steps on a single command. A sequence of steps that the machine does on a single command is called a <u>subroutine</u>. The particular subroutines that we are now going to talk about are deductive shortcuts that are often called <u>derived rules of inference</u>. Our first such derived rule is known as Conjunctive Inference. For future reference we provide the following statement of this rule:

Conjunctive Inference: If $\mathcal{S}$ and $\mathcal{T}$ are well formed formulas which occur as two steps in a demonstration, then $\mathcal{S} \wedge \mathcal{T}$ (or $\mathcal{T} \wedge \mathcal{S}$) may be inferred as a later step in that demonstration. Stated symbolically,

$$\mathfrak{S}, \mathfrak{T} \vdash \mathfrak{S} \wedge \mathfrak{T}$$

$$\mathfrak{S}, \mathfrak{T} \vdash \mathfrak{T} \wedge \mathfrak{S}.$$

Suppose that the machine has printed $\mathfrak{S}$ and $\mathfrak{T}$ as For 1 and For 2, respectively. To get the machine to perform Conjunctive Inference on $\mathfrak{S}$ and $\mathfrak{T}$ and obtain $\mathfrak{S} \wedge \mathfrak{T}$ we give the order

CI, For 1, For 2.

On receiving this command the machine prints three steps

$$\mathfrak{S} \Rightarrow [\mathfrak{T} \Rightarrow [\mathfrak{S} \wedge \mathfrak{T}]]$$

$$\mathfrak{T} \Rightarrow [\mathfrak{S} \wedge \mathfrak{T}]$$

$$\mathfrak{S} \wedge \mathfrak{T}$$

(To obtain $\mathfrak{T} \wedge \mathfrak{S}$ we would give the order " CI, For 2, For 1 ", and three similar steps would be printed out.) Although the machine prints three steps it is designed so that it numbers only the last formula. That is, if $\mathfrak{S}$ and $\mathfrak{T}$ are For 1 and For 2, respectively, and the third order given the machine is " CI, For 1, For 2 ", the machine will identify $\mathfrak{S} \wedge \mathfrak{T}$ as For 3.

1) $\mathfrak{S}$

2) $\mathfrak{T}$

$\mathfrak{S} \Rightarrow [\mathfrak{T} \Rightarrow [\mathfrak{S} \wedge \mathfrak{T}]]$

$\mathfrak{T} \Rightarrow [\mathfrak{S} \wedge \mathfrak{T}]$

3) $\mathfrak{S} \wedge \mathfrak{T}$

The reason for desinging the machine this way is that it enables us to write only an outline of a demonstration and still stay in communication with the machine in such a way that, based on our outline, we can write a program that will cause the machine to print the full demonstration. For example, in order to get the machine to print a demonstration showing that $P, Q, [P \wedge Q] \Rightarrow R \vdash R$ we begin with an outline:

Demonstration Outline		Program Outline
1)	P	Hyp
2)	Q	Hyp
3)	$P \wedge Q$	CI, For 1, For 2
4)	$[P \wedge Q] \Rightarrow R$	Hyp
5)	R	MP, For 4, For 3

We then give the machine the input storage order

Hyp, P, Q, $[P \wedge Q] \Rightarrow R$.

followed by the program

Pr, Hyp 1

Pr, Hyp 2

CI, For 1, For 2

Pr, Hyp 3

MP, For 4, For 3 .

Then the machine printout will be

1) P

2) Q

$P \Rightarrow [Q \Rightarrow [P \wedge Q]]$

$Q \Rightarrow [P \wedge Q]$

3) $P \wedge Q$

4) $[P \wedge Q] \Rightarrow R$

5) R

A second subroutine which may be identified from the examples of the last section is the "reverse" of Conjunctive Inference and it comes in two forms.

Conjunctive Simplification: If $\mathcal{S}$ and $\mathcal{T}$ are well formed formulas and $\mathcal{S} \wedge \mathcal{T}$ occurs as a step in a demonstration, then either

(i) $\mathcal{S}$ (Right[*] Conjunctive Simplification)

or

(ii) $\mathcal{T}$ (Left Conjunctive Simplification)

(or both) may be inferred as a later step (or steps) in that demonstration.

Stated symbolically:

(i) $\mathcal{S} \wedge \mathcal{T} \vdash \mathcal{S}$

(ii) $\mathcal{S} \wedge \mathcal{T} \vdash \mathcal{T}$.

These inferences are made as follows: To conclude $\mathcal{S}$ we use the tautology

$$[\mathcal{S} \wedge \mathcal{T}] \Rightarrow \mathcal{S}$$

and Modus Ponens; to conclude $\mathcal{T}$ we use

$$[\mathcal{S} \wedge \mathcal{T}] \Rightarrow \mathcal{T}$$

and Modus Ponens.

Suppose, for example, that the machine has printed $\mathcal{S} \wedge \mathcal{T}$ as For 1. For Right Conjunctive Simplification we give the order

RCS, For 1.

On receiving this order the machine examines For 1 to check that it is a conjunction. If it is, the machine identifies the first component, $\mathcal{S}$, of the conjunction and prints

$$[\mathcal{S} \wedge \mathcal{T}] \Rightarrow \mathcal{S}$$

$$\mathcal{S}$$

Thus if "RCS, For 1" is the second order the machine receives, it will identify $\mathcal{S}$ as For 2.

---

[*] One might think that, since $\mathcal{S}$ is the left hand component of $\mathcal{S} \wedge \mathcal{T}$, it would be more logical to call this "Left Conjunctive Simplification". However, our title is justified if one views the situation in the following perspective: the simplification consists in the removal of one of the components of $\mathcal{S} \wedge \mathcal{T}$, and to obtain $\mathcal{S}$ one has to remove the right hand component.

For Left Conjunctive Simplification the order is

LCS, For __.

Before we turn to the problem of designing other subroutines, let us illustrate the use of the two routines we have just introduced.

Example 1: Show that P, Q $\vdash$ ~[~P ∨ ~Q].

	Demonstration Outline	Program Outline
1)	P	Hyp
2)	Q	Hyp
3)	P ∧ Q	CI, For 1, For 2
4)	[P ∧ Q] ⇒ ~[~P ∨ ~Q]	Taut
5)	~[~P ∨ ~Q]	MP, For 4, For 3.

Example 2: Show that P ∧ Q, P ⇒ [R ∨ S] $\vdash$ R ∨ S.

	Demonstration Outline	Program Outline
1)	P ∧ Q	Hyp
2)	P	RCS, For 1
3)	P ⇒ [R ∨ S]	Hyp
4)	R ∨ S	MP, For 3, For 2.

Example 3: Show that P, Q, R ∧ S, [P ∧ Q] ⇒ [R ⇒ T] $\vdash$ T.

	Demonstration Outline	Program Outline
1)	P	Hyp
2)	Q	Hyp
3)	P ∧ Q	CI, For 1, For 2
4)	[P ∧ Q] ⇒ [R ⇒ T]	Hyp
5)	R ⇒ T	MP, For 4, For 3
6)	R ∧ S	Hyp

7)	R	RCS, For 6
8)	T	MP, For 5, For 7.

Example 4: Show that $P \wedge R,\ R \Leftrightarrow S \vdash R \vee S$.

	Demonstration Outline	Program Outline
1)	$R \Leftrightarrow S$	Hyp
2)	$R \Rightarrow S$	RCS, For 1 ⁎
3)	$P \wedge R$	Hyp
4)	R	LCS, For 3
5)	S	MP, For 2, For 4
6)	$R \wedge S$	CI, For 4, For 5
7)	$[R \wedge S] \Rightarrow [R \vee S]$	Taut
8)	$R \vee S$	MP, For 7, For 6.

This 8-step outline will lead to a 12-step demonstration. But here is a simpler demonstration of the same result:

	Demonstration Outline	Program Outline
1)	$P \wedge R$	Hyp
2)	R	LCS, For 1
3)	$R \Rightarrow [R \vee S]$	Taut
4)	$R \vee S$	MP, For 3, For 2

This four-step outline will produce a five-step demonstration. There are several observations to be made here. First we see that demonstrations are not unique for here we have two different demonstrations showing that "$R \vee S$" follows logically from "$P \wedge R$" and "$R \Leftrightarrow S$". Second, while the first demonstration uses the hypothesis "$R \Leftrightarrow S$" the second does not. Thus the second demonstration shows that "$R \vee S$" follows logically from

---

⁎ Don't forget: "$R \Leftrightarrow S$" is a shorthand for "$[R \Rightarrow S] \wedge [S \Rightarrow R]$".

" $P \wedge R$ " alone. Since " $R \vee S$ " follows from " $P \wedge R$ " it also follows from any collection of formulas that contains " $P \wedge R$ ". This is a particular case of a general result about unnecessary hypotheses which may be stated thus:

Principle of Unnecessary Hypotheses: If $\mathcal{S}_1, \ldots, \mathcal{S}_n, \mathcal{R}$ are well formed formulas, if $\mathcal{S}_1, \ldots, \mathcal{S}_n \vdash \mathcal{R}$ and if $\mathcal{T}_1, \ldots, \mathcal{T}_m$ is any collection of well formed formulas, then $\mathcal{S}_1, \ldots, \mathcal{S}_n, \mathcal{T}_1, \ldots, \mathcal{T}_m \vdash \mathcal{R}$.

As we study mathematical arguments we will want to be on the lookout for unnecessary hypotheses. If a hypothesis is not used then the result being proved can be improved by deleting the superfluous hypothesis. Even if a hypothesis is used it may still be possible to prove the result without it. This has been illustrated above by Example 4.

We give one final example and then we will provide some problems for you to solve.

Example 5: Show that $R \wedge [P \vee {}^{\sim}S], [P \vee {}^{\sim}S] \Rightarrow [[Q \Rightarrow T] \wedge S] \vdash [Q \Rightarrow T] \wedge R$.

	Demonstration Outline	Program Outline
1)	$R \wedge [P \vee {}^{\sim}S]$	Hyp
2)	$P \vee {}^{\sim}S$	LCS, For 1
3)	$[P \vee {}^{\sim}S] \Rightarrow [[Q \Rightarrow T] \wedge S]$	Hyp
4)	$[Q \Rightarrow T] \wedge S$	MP, For 3, For 2
5)	$Q \Rightarrow T$	RCS, For 4
6)	R	RCS, For 1
7)	$[Q \Rightarrow T] \wedge R$	CI, For 5, For 6.

PROBLEM SET 7

In Problems 1 - 4, complete the program outline of the demonstration.

1. Show that $\sim Q \Rightarrow \sim P$, $\sim Q \wedge R \vdash \sim P$.

Demonstration Outline		Program Outline
1)	$\sim Q \wedge R$	
2)	$\sim Q$	
3)	$\sim Q \Rightarrow \sim P$	
4)	$\sim P$	

2. Show that $R \Rightarrow S$, $R \Rightarrow P$, $R \vdash P \wedge S$.

Demonstration Outline		Program Outline
1)	$R \Rightarrow S$	
2)	R	
3)	S	
4)	$R \Rightarrow P$	
5)	P	
6)	$P \wedge S$	

3. Show that $S \wedge P$, $P \Rightarrow Q$, $Q \Rightarrow R \vdash R$.

Demonstration Outline		Program Outline
1)	$P \Rightarrow Q$	
2)	$Q \Rightarrow R$	
3)	$[P \Rightarrow Q] \wedge [Q \Rightarrow R]$	
4)	$[[P \Rightarrow Q] \wedge [Q \Rightarrow R]] \Rightarrow [P \Rightarrow R]$	
5)	$P \Rightarrow R$	
6)	$S \wedge P$	
7)	P	
8)	R	

4. Show that $P, P \Rightarrow Q \vdash P \wedge Q$.

Demonstration Outline	Program Outline
1) P	
2) $P \Rightarrow Q$	
3) Q	
4) $P \wedge Q$	

In Problems 5 - 12, outline a demonstration of the given result and provide a program outline for that demonstration.

5. Show that $R \Rightarrow S,\ R \wedge Q \vdash R \wedge S$.

6. Show that $\sim[P \wedge R] \Rightarrow \sim S,\ \sim S \Rightarrow Q,\ \sim[P \wedge R] \vdash \sim S \wedge Q$.

7. Show that $[P \vee R] \Rightarrow Q,\ P \vdash P \wedge Q$.

8. Show that $R \wedge \sim Q,\ P \Rightarrow Q \vdash R \wedge \sim P$.

   (Hint: Use the tautology "$[\sim Q \wedge [P \Rightarrow Q]] \Rightarrow \sim P$".)

9. Show that $P \wedge Q,\ P \Rightarrow R \vdash R \vee S$.

10. Show that $Q,\ [P \Rightarrow Q] \Rightarrow [R \wedge S] \vdash S$.

11. Show that $\sim Q,\ P \Rightarrow Q \vdash \sim P$.

12. Show that $R \Leftrightarrow S,\ R \vdash S$.

13. Each of the following involves one or more unnecessary hypothesis. In each case, identify as many unnecessary hypotheses as you can and then outline an appropriate demonstration using your reduced set of hypotheses. Provide a program outline for each demonstration outline.

   a) $P,\ Q \vdash P \Rightarrow Q$

   b) $R \Leftrightarrow S,\ P \vee S,\ R \Rightarrow \sim P,\ \sim[S \wedge Q] \vdash R \Rightarrow \sim Q$

   c) $\sim P \vee R,\ [\sim P \vee R] \Rightarrow Q,\ [Q \wedge P] \Rightarrow R,\ P \vdash R$

   d) $P \wedge Q,\ [P \vee Q] \Rightarrow \sim S,\ Q \Rightarrow \sim T,\ Q \vdash \sim[[S \wedge T] \vee \sim Q]$

Review: In each part of the following, use De Morgan's laws to find a formula equivalent to the given formula.

14.

a) $\sim[P \wedge Q]$

b) $\sim R \vee \sim T$

c) $\sim[\sim R \wedge S]$

d) $\sim R \vee \sim S$

e) $\sim P \wedge \sim Q$

f) $\sim[\sim R] \wedge \sim S$

g) $\sim[P \vee \sim Q]$

h) $\sim[\sim R \wedge \sim S]$

i) $\sim P \vee \sim[\sim Q]$

j) $P \wedge Q$

- o O o -

Aftermath...

Taxi driver's analysis of what's wrong with education in our country today: "The trouble is 80% of the kids can't read; the other 30% don't know their mathematics."

- o O o -

2.3 Contrapositive Inference and Modus Tollens

Let us now return to the problem of designing useful subroutines. Thus far we have made only three:

$$\left.\begin{array}{l} \mathcal{S}, \mathcal{T} \vdash \mathcal{S} \wedge \mathcal{T} \\ \mathcal{S}, \mathcal{T} \vdash \mathcal{T} \wedge \mathcal{S} \end{array}\right\} \quad \text{Conjunctive Inference}$$

$\mathcal{S} \wedge \mathcal{T} \vdash \mathcal{S}$ Right Conjunctive Simplification

$\mathcal{S} \wedge \mathcal{T} \vdash \mathcal{T}$ Left Conjunctive Simplification

To this list we now wish to add two new subroutines:

$\mathcal{S} \Rightarrow \mathcal{T} \vdash \sim\mathcal{T} \Rightarrow \sim\mathcal{S}$ Contrapositive Inference

$\mathcal{S} \Rightarrow \mathcal{T}, \sim\mathcal{T} \vdash \sim\mathcal{S}$ Modus Tollens[†]

As an example of Contrapositive Inference let us assume the machine has printed " $P \Rightarrow Q$ " as For 1. On receiving the order

CPI, For 1

[†] In classical logic the full name for this derived rule of inference is "Modus Tollendo Tollens", which may literally be translated as "The rule (modus) that denies (tollens) by denying (tollendo)"; the meaning being that, given the conditional $\mathcal{S} \Rightarrow \mathcal{T}$, if you deny (i.e., affirm the negation of) the consequent, $\mathcal{T}$, then you may also deny the antecedent, $\mathcal{S}$.

the machine checks For 1 to make sure it is a conditional, identifies its antecedent "P" and its consequent "Q", and prints

$$[P \Rightarrow Q] \Rightarrow [{\sim}Q \Rightarrow {\sim}P]$$

$${\sim}Q \Rightarrow {\sim}P$$

Here the first formula is a tautology and the second follows from the first and For 1 by Modus Ponens. Thus we see that applying the Contrapositive Inference subroutine causes the machine to perform a sequence of two steps at a single command.

For Modus Tollens the order is

MT, For 1, For 2

if " $P \Rightarrow Q$ " and " ${\sim}Q$ " have been printed as For 1 and For 2, respectively. The machine first examines For 1 and For 2 to check that they are suitable for applying the rule. Finding that they are, the machine prints

$$[P \Rightarrow Q] \Rightarrow [{\sim}Q \Rightarrow {\sim}P]$$

$${\sim}Q \Rightarrow {\sim}P$$

$${\sim}P$$

Hence the Modus Tollens subroutine causes the performance of three steps at a single command. As in the case of Modus Ponens, the order in which you mention the formulas to which you want the machine to apply Modus Tollens is important. The first formula mentioned must be the conditional, and the second formula mentioned must be the negation of the consequent of that conditional. Once again, it would be possible to modify our machine so that it would accept the formulas in either order, but we will not concern ourselves with that at this point.

As you will have noticed, the Contrapositive Inference and Modus Tollens subroutines are closely related. Indeed, the first two steps of the three that result from a Modus Tollens command are identical with the response to a Contrapositive Inference command. This means that every demonstration which at some stage makes use of Modus Tollens could be rewritten, by the addition of an appropriate number (i.e., the number of times Modus Tollens was used) of single steps, as a demonstration making use of Contrapositive Inference. In other words, every use of Modus Tollens really includes a use of Contrapositive Inference.

So that you will be able to see how it is used in practice, we now provide some examples of the use of Modus Tollens. Of course, in light of the preceding discussion, these will also be (disguised) examples of the use of Contrapositive Inference.

Example 1: Show that $\tilde{}P \Rightarrow Q$, $Q \Rightarrow R$, $\tilde{}R \vdash P$.

	Demonstration Outline	Program Outline
1)	$\tilde{}P \Rightarrow Q$	Hyp
2)	$Q \Rightarrow R$	Hyp
3)	$\tilde{}R$	Hyp
4)	$\tilde{}Q$	MT, For 2, For 3
5)	$\tilde{}[\tilde{}P]$	MT, For 1, For 4
6)	$\tilde{}[\tilde{}P] \Rightarrow P$	Taut
7)	P	MP, For 6, For 5

Example 2: Show that $R \wedge \tilde{}S$, $\tilde{}P \Rightarrow S \vdash R \wedge P$.

	Demonstration Outline	Program Outline
1)	$R \wedge \tilde{}S$	Hyp
2)	R	RCS, For 1
3)	$\tilde{}S$	LCS, For 1
4)	$\tilde{}P \Rightarrow S$	Hyp
5)	$\tilde{}[\tilde{}P]$	MT, For 4, For 3
6)	$\tilde{}[\tilde{}P] \Rightarrow P$	Taut
7)	P	MP, For 6, For 5
8)	$R \wedge P$	CI, For 2, For 7

Example 3: Show that R, $R \Rightarrow [P \wedge Q]$, $[P \wedge S] \Rightarrow \tilde{}P \vdash \tilde{}P \vee \tilde{}S$.

	Demonstration Outline	Program Outline
1)	R	Hyp
2)	$R \Rightarrow [P \wedge Q]$	Hyp

3)	$P \wedge Q$	MP, For 2, For 1
4)	P	RCS, For 3
5)	$[P \wedge S] \Rightarrow {}^{\sim}P$	Hyp
6)	$P \Rightarrow {}^{\sim}[{}^{\sim}P]$	Taut
7)	${}^{\sim}[{}^{\sim}P]$	MP, For 6, For 4
8)	${}^{\sim}[P \wedge S]$	MT, For 5, For 7
9)	${}^{\sim}[P \wedge S] \Rightarrow [{}^{\sim}P \vee {}^{\sim}S]$	Taut
10)	${}^{\sim}P \vee {}^{\sim}S$	MP, For 9, For 8

PROBLEM SET 8

In Problems 1 - 3, complete the program outline of the demonstration.

1. Show that $[P \wedge Q] \Rightarrow R,\ {}^{\sim}R,\ P \vdash {}^{\sim}Q$.

	Demonstration Outline	Program Outline
1)	$[P \wedge Q] \Rightarrow R$	
2)	${}^{\sim}R$	
3)	${}^{\sim}[P \wedge Q]$	
4)	${}^{\sim}[P \wedge Q] \Rightarrow [{}^{\sim}P \vee {}^{\sim}Q]$	
5)	${}^{\sim}P \vee {}^{\sim}Q$	
6)	$[{}^{\sim}P \vee {}^{\sim}Q] \Rightarrow [P \Rightarrow {}^{\sim}Q]$	
7)	$P \Rightarrow {}^{\sim}Q$	
8)	P	
9)	${}^{\sim}Q$	

2. Show that $Q \Rightarrow S,\ {}^{\sim}S \vee R,\ {}^{\sim}R \vdash {}^{\sim}Q$.

	Demonstration Outline	Program Outline
1)	${}^{\sim}S \vee R$	
2)	$[{}^{\sim}S \vee R] \Rightarrow [S \Rightarrow R]$	
3)	$S \Rightarrow R$	

4) $\sim R$

5) $\sim S$

6) $Q \Rightarrow S$

7) $\sim Q$

3) Show that $\sim P,\ [R \vee Q] \Rightarrow P,\ S \Rightarrow [Q \vee R] \vdash \sim[P \vee S]$.

Demonstration Outline		Program Outline
1)	$\sim P$	
2)	$[R \vee Q] \Rightarrow P$	
3)	$\sim[R \vee Q]$	
4)	$S \Rightarrow [Q \vee R]$	
5)	$[S \Rightarrow [Q \vee R]] \Rightarrow [S \Rightarrow [R \vee Q]]$	
6)	$S \Rightarrow [R \vee Q]$	
7)	$\sim S$	
8)	$\sim P \wedge \sim S$	
9)	$[\sim P \wedge \sim S] \Rightarrow \sim[P \vee S]$	
10)	$\sim[P \vee S]$	

In Problems 4 - 14, outline a demonstration of the given result and provide a program outline for that demonstration.

4. Show that $\sim Q,\ P \Rightarrow Q,\ \sim P \Rightarrow R \vdash R$.

5. Show that $R \Rightarrow S,\ \sim R \Rightarrow \sim[\sim P],\ \sim S \vdash P$.

6. Show that $P \wedge \sim Q,\ \sim R \Rightarrow Q \vdash P \wedge R$.

7. Show that $P \Rightarrow [R \vee S],\ [R \vee S] \Rightarrow Q,\ P \vdash Q$.

8. Show that $[\sim P \wedge S] \Rightarrow R,\ \sim P \Rightarrow S,\ \sim P \vdash \sim[P \vee \sim R]$.

9. Show that $[P \vee Q] \vee R \vdash R \vee [P \vee Q]$.

10. Show that $S \Rightarrow \sim Q,\ Q,\ \sim S \Rightarrow [T \wedge U] \vdash [T \wedge Q] \wedge U$.

11. Show that $\sim[P \Rightarrow Q],\ R \Rightarrow Q \vdash \sim R$.

12. Show that $P \Leftrightarrow Q,\ Q \Leftrightarrow R \vdash P \Leftrightarrow R$.

13. Show that $P \vee Q,\ Q \Rightarrow [S \wedge {\sim}T],\ P \Rightarrow U,\ {\sim}U \vdash {\sim}T \wedge S$.

14. Show that $P \wedge Q,\ {\sim}R \Rightarrow {\sim}[S \wedge T],\ [Q \wedge P] \Rightarrow [T \wedge S] \vdash P \wedge R$.

Review: In each part of the following, use the distributive laws

$$[\mathcal{S} \wedge [\mathcal{T} \vee \mathcal{U}]] \Leftrightarrow [[\mathcal{S} \wedge \mathcal{T}] \vee [\mathcal{S} \wedge \mathcal{U}]]$$

$$[\mathcal{S} \vee [\mathcal{T} \wedge \mathcal{U}]] \Leftrightarrow [[\mathcal{S} \vee \mathcal{T}] \wedge [\mathcal{S} \vee \mathcal{U}]]$$

to find a formula equivalent to the given formula.

15. a) $[P \wedge S] \vee [P \wedge R]$

b) $Q \wedge [P \vee R]$

c) ${\sim}R \vee [{\sim}P \wedge S]$

d) $[[S \vee Q] \wedge R] \vee [[S \vee Q] \wedge P]$

e) $[{\sim}Q \vee R] \wedge [{\sim}Q \vee P]$

f) ${\sim}S \wedge [Q \vee {\sim}S]$

g) $[[P \wedge R] \vee S] \wedge [[P \wedge R] \vee Q]$

h) $[{\sim}P \wedge Q] \wedge [R \vee S]$

- o O o -

Aftermath...

There is an exception to every rule.
That statement itself is a rule.
If it is a rule, there is an exception to it.
If it has an exception, there's <u>not</u> an exception to every rule.
So, if the statement is true, it's false.

- o O o -

2.4 Syllogistic Inference and Inference by Cases

In this section we present two more subroutines:

$$\mathcal{S} \Rightarrow \mathcal{T},\ \mathcal{T} \Rightarrow \mathcal{U} \vdash \mathcal{S} \Rightarrow \mathcal{U} \qquad \text{Syllogistic Inference}$$

$$\left.\begin{array}{l} \mathcal{S} \Rightarrow \mathcal{T},\ \mathcal{U} \Rightarrow \mathcal{T} \vdash [\mathcal{S} \vee \mathcal{U}] \Rightarrow \mathcal{T} \\ \mathcal{S} \Rightarrow \mathcal{T},\ \mathcal{U} \Rightarrow \mathcal{T} \vdash [\mathcal{U} \vee \mathcal{S}] \Rightarrow \mathcal{T} \end{array}\right\} \qquad \text{Inference by Cases}$$

Suppose, for example, the machine has printed " $P \Rightarrow Q$ " and " $Q \Rightarrow R$ " as For 1 and For 2, respectively. The order that will cause the machine to perform Syllogistic Inference on these two formulas is

SI, For 1, For 2.

On receiving this order the machine checks For 1 and For 2 to make sure that they are suitable. On finding that they are it prints

$$[P \Rightarrow Q] \Rightarrow [[Q \Rightarrow R] \Rightarrow [P \Rightarrow R]]$$

$$[Q \Rightarrow R] \Rightarrow [P \Rightarrow R]$$

$$P \Rightarrow R$$

Here the first formula is a tautology and the second and third formulas follow from the first, For 1, and For 2 by Modus Ponens. Notice that in this subroutine, as in some of the ones we met previously, you have to be careful to mention the two formulas to which you want the machine to apply Syllogistic Inference in the correct order.

Similarly, if the machine has printed " $P \Rightarrow Q$ " and " $R \Rightarrow Q$ " as For 1 and For 2, respectively, the order that will cause the machine to make an Inference by Cases from For 1 and For 2 and obtain " $[P \vee R] \Rightarrow Q$ " is

IC, For 1, For 2.

On receiving this order the machine prints

$$[P \Rightarrow Q] \Rightarrow [[R \Rightarrow Q] \Rightarrow [[P \vee R] \Rightarrow Q]]$$

$$[R \Rightarrow Q] \Rightarrow [[P \vee R] \Rightarrow Q]$$

$$[P \vee R] \Rightarrow Q$$

Again the first formula is a tautology and the second and third follow from the first, For 1, and For 2 by Modus Ponens. The command in order to obtain " $[R \vee P] \Rightarrow Q$ " would be

" IC, For 2, For 1 ", on receipt of which the machine would print a similar sequence of three formulas.

We now provide some examples which show Syllogistic Inference and Inference by Cases in use.

Example 1: Show that $P \Rightarrow Q,\ R \Rightarrow {}^{\sim}Q \vdash P \Rightarrow {}^{\sim}R$.

	Demonstration Outline	Program Outline
1)	$P \Rightarrow Q$	Hyp
2)	$R \Rightarrow {}^{\sim}Q$	Hyp
3)	${}^{\sim}[{}^{\sim}Q] \Rightarrow {}^{\sim}R$	CPI, For 2
4)	$Q \Rightarrow {}^{\sim}[{}^{\sim}Q]$	Taut
5)	$Q \Rightarrow {}^{\sim}R$	SI, For 4, For 3
6)	$P \Rightarrow {}^{\sim}R$	SI, For 1, For 5

Example 2: Show that ${}^{\sim}P \Rightarrow Q,\ Q \Rightarrow R,\ {}^{\sim}R \vdash P$.

	Demonstration Outline	Program Outline
1)	${}^{\sim}P \Rightarrow Q$	Hyp
2)	$Q \Rightarrow R$	Hyp
3)	${}^{\sim}P \Rightarrow R$	SI, For 1, For 2
4)	${}^{\sim}R$	Hyp
5)	${}^{\sim}[{}^{\sim}P]$	MT, For 3, For 4
6)	${}^{\sim}[{}^{\sim}P] \Rightarrow P$	Taut
7)	P	MP, For 6, For 5

Example 3: Show that $P \Rightarrow [Q \vee R],\ Q \Rightarrow S,\ R \Rightarrow S \vdash P \Rightarrow S$.

	Demonstration Outline	Program Outline
1)	$P \Rightarrow [Q \vee R]$	Hyp
2)	$Q \Rightarrow S$	Hyp
3)	$R \Rightarrow S$	Hyp

4)	$[Q \vee R] \Rightarrow S$	IC, For 2, For 3
5)	$P \Rightarrow S$	SI, For 1, For 4

PROBLEM SET 9

In Problems 1 - 4, complete the program outline of the demonstration.

1. Show that $Q \Rightarrow S,\ R \Rightarrow \sim S,\ P \Rightarrow R \vdash P \Rightarrow \sim Q$.

	Demonstration Outline	Program Outline
1)	$P \Rightarrow R$	
2)	$R \Rightarrow \sim S$	
3)	$P \Rightarrow \sim S$	
4)	$Q \Rightarrow S$	
5)	$\sim S \Rightarrow \sim Q$	
6)	$P \Rightarrow \sim Q$	

2. Show that $S \Rightarrow Q,\ [R \Rightarrow Q] \Rightarrow [P \wedge T],\ R \Rightarrow S \vdash T$.

	Demonstration Outline	Program Outline
1)	$R \Rightarrow S$	
2)	$S \Rightarrow Q$	
3)	$R \Rightarrow Q$	
4)	$[R \Rightarrow Q] \Rightarrow [P \wedge T]$	
5)	$P \wedge T$	
6)	T	

3. Show that $Q \Rightarrow S,\ \sim R \Rightarrow Q,\ R \Rightarrow S \vdash S$.

	Demonstration Outline	Program Outline
1)	$\sim R \Rightarrow Q$	
2)	$[\sim R \Rightarrow Q] \Rightarrow [R \vee Q]$	
3)	$R \vee Q$	
4)	$R \Rightarrow S$	

5) $Q \Rightarrow S$

6) $[R \vee Q] \Rightarrow S$

7) S

4. Show that $\sim P,\ Q \Rightarrow P,\ R \Rightarrow P \vdash \sim[R \vee Q]$.

Demonstration Outline — Program Outline

1) $Q \Rightarrow P$

2) $R \Rightarrow P$

3) $[R \vee Q] \Rightarrow P$

4) $\sim P$

5) $\sim[R \vee Q]$

In Problems 5 - 16, outline a demonstration of the given result and provide a program outline for that demonstration.

5. Show that $[Q \Rightarrow R] \wedge P,\ [Q \Rightarrow R] \Rightarrow \sim T \vdash \sim T$.

6. Show that $Q \Rightarrow R,\ \sim P \Rightarrow Q,\ \sim R \vdash P$.

7. Show that $\sim R \Rightarrow S,\ S \Rightarrow [P \wedge Q],\ R \Rightarrow T,\ \sim T \vdash Q$.

8. Show that $Q \Rightarrow S,\ \sim S,\ R \Rightarrow S \vdash \sim Q \wedge \sim R$.

9. Show that $R \vee S,\ P,\ S \Rightarrow [Q \wedge T],\ P \Rightarrow [R \Rightarrow [Q \wedge T]] \vdash Q$.

10. Show that $S \wedge R,\ Q \Rightarrow T,\ R \Rightarrow \sim T \vdash \sim Q \wedge S$.

11. Show that $\sim S \Rightarrow \sim[P \vee \sim T],\ T \Rightarrow [Q \wedge R],\ \sim S \vdash R \wedge Q$.

12. Show that $P \Rightarrow Q,\ R,\ R \Rightarrow [Q \Rightarrow P] \vdash P \Leftrightarrow Q$.

13. Show that $\sim[\sim P \wedge \sim Q],\ S \Rightarrow \sim Q,\ \sim P \vee S \vdash Q \Leftrightarrow \sim P$.

14. Show that $P \Rightarrow Q,\ R \Rightarrow S,\ Q \Rightarrow R \vdash P \Rightarrow S$.

15. Show that $P,\ P \Rightarrow Q,\ Q \Rightarrow R \vdash R$.

16. Show that $\sim P \Rightarrow Q,\ [R \Rightarrow Q] \Rightarrow S,\ \sim S \vee T,\ R \Rightarrow \sim P \vdash T \vee V$.

17. In each of the following the last sentence is a logical consequence of the preceding statements. Show that this is so by identifying well formed formulas which are patterns for each sentence, outlining a demonstration, and providing a program outline for the demonstration.

Example: If today is Monday, then yesterday was not a school day. Yesterday and the day before were school days. So, today isn't Monday.

Take M: Today is Monday   Y: Yesterday was a school day
B: The day before yesterday was a school day.

The argument then becomes: $M \Rightarrow {}^{\sim}Y,\ Y \wedge B \vdash {}^{\sim}M$.

	Demonstration Outline	Program Outline
1)	$Y \wedge B$	Hyp
2)	Y	RCS, For 1
3)	$M \Rightarrow {}^{\sim}Y$	Hyp
4)	$Y \Rightarrow {}^{\sim}[{}^{\sim}Y]$	Taut
5)	${}^{\sim}[{}^{\sim}Y]$	MP, For 4, For 2
6)	${}^{\sim}M$	MT, For 3, For 5

a) I can't go to Mary's party and also go to Jim's. If I go to Jim's, Mary will be upset. I won't upset Mary, but I will go to one of the parties. So, I will go to Mary's.

M: I go to Mary's party   J: I go to Jim's party
U: Mary will be upset

The argument then becomes: ${}^{\sim}[M \wedge J],\ J \Rightarrow U,\ {}^{\sim}U \wedge [M \vee J] \vdash M$.

b) If Bill doesn't play baseball, then Pete will be the shortstop. If Mike doesn't pitch, then Pete won't play shortstop. Therefore, if Bill doesn't play, Mike will pitch.

c) If mathematics is important as an experience in thinking, then we should study it. If it is important as a tool for other subjects, we should also study it. Mathematics is important either as an experience in thinking or as a tool for other subjects. So, we should study mathematics.

Review: In each part of the following, find a disjunction equivalent to the given formula.

18. a) $P \Rightarrow Q$

b) $\sim P \Rightarrow Q$

c) $P \Rightarrow \sim Q$

- o O o -

2.5 Subroutines Involving the Biconditional; The General Substitution Principle.

Recall that in Chapter 1 we introduced a way to abbreviate formulas such as

(*) $[\mathcal{S} \Rightarrow \mathcal{T}] \wedge [\mathcal{T} \Rightarrow \mathcal{S}]$

where $\mathcal{S}$ and $\mathcal{T}$ are well formed formulas. We did this by means of the biconditional "$\Leftrightarrow$"; formula (*) may be written in the shorthand form

$[\mathcal{S} \Leftrightarrow \mathcal{T}]$.

As it happens, formulas such as formula (*) play a very special and significant role in later work. That is why it is so convenient to have a shorthand way of writing such formulas. The object of this section is to introduce several subroutines which involve the biconditional and which will save us some effort in our later work.

Modus Ponens assures us that, if $\mathcal{S}$ and $\mathcal{T}$ are well formed formulas, we can infer $\mathcal{T}$ from $\mathcal{S}$ and $\mathcal{S} \Rightarrow \mathcal{T}$. But it is also the case that we can conclude $\mathcal{T}$ from $\mathcal{S}$ and $\mathcal{S} \Leftrightarrow \mathcal{T}$ as the following demonstration outline shows.

1)	$\mathcal{S}$	Hyp
2)	$\mathcal{S} \Leftrightarrow \mathcal{T}$	Hyp
3)	$\mathcal{S} \Rightarrow \mathcal{T}$	RCS, For 2
4)	$\mathcal{T}$	MP, For 3, For 1

We make use of this to build into our machine a new subroutine called Modus Ponens for the Biconditional. Suppose, for example, the machine has printed "P" and "$P \Leftrightarrow Q$" as For 1 and For 2, respectively. The order that will cause the machine to apply Modus Ponens for the Biconditional to these two formulas is

MPB, For 2, For 1.

On receiving this order the machine checks For 1 and For 2 to see if they are suitable.

Since they are, it then prints

$P \Rightarrow Q$

Q

Notice that you have to be careful to mention the biconditional first when giving the order to the machine.*

Example 1: Show that $R,\ R \Leftrightarrow [P \wedge Q],\ [P \wedge S] \Rightarrow {\sim}P \vdash {\sim}P \vee {\sim}S$.

	Demonstration Outline	Program Outline
1)	R	Hyp
2)	$R \Leftrightarrow [P \wedge Q]$	Hyp
3)	$P \wedge Q$	MPB, For 2, For 1
4)	P	RCS, For 3
5)	$P \Rightarrow {\sim}[{\sim}P]$	Taut
6)	${\sim}[{\sim}P]$	MP, For 5, For 4
7)	$[P \wedge S] \Rightarrow {\sim}P$	Hyp
8)	${\sim}[P \wedge S]$	MT, For 7, For 6
9)	${\sim}[P \wedge S] \Leftrightarrow [{\sim}P \vee {\sim}S]$	Taut
10)	${\sim}P \vee {\sim}S$	MPB, For 9, For 8

Example 2: Show that $[Q \Rightarrow R] \Leftrightarrow [P \Rightarrow R],\ P \Rightarrow R,\ {\sim}R \vdash {\sim}P \wedge {\sim}Q$.

	Demonstration Outline	Program Outline
1)	$[Q \Rightarrow R] \Leftrightarrow [P \Rightarrow R]$	Hyp
2)	$P \Rightarrow R$	Hyp
3)	$Q \Rightarrow R$	MPB, For 1, For 2
4)	$[P \vee Q] \Rightarrow R$	IC, For 2, For 3
5)	${\sim}R$	Hyp
6)	${\sim}[P \vee Q]$	MT, For 4, For 5

* Modus Ponens for the Biconditional can also be applied in the other order. That is, if "Q" and "$P \Leftrightarrow Q$" are For 1 and For 2, respectively, then the MPB subroutine will cause the machine to print "$Q \Rightarrow P$" and then "P".

7)	$\sim[P \vee Q] \Leftrightarrow [\sim P \wedge \sim Q]$	Taut
8)	$\sim P \wedge \sim Q$	MPB, For 7, For 6

Our next subroutine relies on the fact that, for well formed formulas $\mathcal{S}$, $\mathcal{T}$, and $\mathcal{U}$,

$$\mathcal{S} \Leftrightarrow \mathcal{T},\ \mathcal{T} \Leftrightarrow \mathcal{U} \vdash \mathcal{S} \Leftrightarrow \mathcal{U}.$$

This result (known as the Transitive Property of the Biconditional) is easily shown, as outlined below.

1)	$\mathcal{S} \Leftrightarrow \mathcal{T}$	Hyp
2)	$\mathcal{T} \Leftrightarrow \mathcal{U}$	Hyp
3)	$\mathcal{S} \Rightarrow \mathcal{T}$	RCS, For 1
4)	$\mathcal{T} \Rightarrow \mathcal{U}$	RCS, For 2
5)	$\mathcal{S} \Rightarrow \mathcal{U}$	SI, For 3, For 4
6)	$\mathcal{T} \Rightarrow \mathcal{S}$	LCS, For 1
7)	$\mathcal{U} \Rightarrow \mathcal{T}$	LCS, For 2
8)	$\mathcal{U} \Rightarrow \mathcal{S}$	SI, For 7, For 6
9)	$\mathcal{S} \Leftrightarrow \mathcal{U}$	CI, For 5, For 8

We build into our machine a new subroutine which enables us to make direct use of this property. With this modification, if the machine has printed " $P \Leftrightarrow Q$ " and " $Q \Leftrightarrow R$ " as For 1 and For 2, respectively, and you give it the order

TPB, For 1, For 2

the machine will print out a whole sequence of steps (similar to formulas (3) - (9) above), ending with " $P \Leftrightarrow R$ ". Notice once again that the order in which you mention the two formulas in the command you give the machine is important.

Example: Show that $P \Leftrightarrow R,\ R \Rightarrow [S \wedge T],\ S \Rightarrow R \vdash P \Leftrightarrow [S \wedge T]$

	Demonstration Outline	Program Outline
1)	$P \Leftrightarrow R$	Hyp
2)	$R \Rightarrow [S \wedge T]$	Hyp
3)	$S \Rightarrow R$	Hyp

4)	$[S \Rightarrow R] \Rightarrow [[S \wedge T] \Rightarrow R]$	Taut
5)	$[S \wedge T] \Rightarrow R$	MP, For 4, For 3
6)	$R \Leftrightarrow [S \wedge T]$	CI, For 2, For 5
7)	$P \Leftrightarrow [S \wedge T]$	TPB, For 1, For 6

Our third new subroutine relies on the <u>Commutative Property of the Biconditional</u>, namely,

$$\mathcal{S} \Leftrightarrow \mathcal{T} \vdash \mathcal{T} \Leftrightarrow \mathcal{S}$$

for well formed formulas $\mathcal{S}$ and $\mathcal{T}$. This may be shown briefly as follows:

1)	$\mathcal{S} \Leftrightarrow \mathcal{T}$	Hyp
2)	$[\mathcal{S} \Leftrightarrow \mathcal{T}] \Rightarrow [\mathcal{T} \Leftrightarrow \mathcal{S}]$	Taut
3)	$\mathcal{T} \Leftrightarrow \mathcal{S}$	MP, For 2, For 1

With a suitable modification of our machine we can make direct use of this property too. Suppose, for example the machine has printed " $P \Leftrightarrow Q$ " as For 1, then the command " CPB, For 1 " will cause it to print

$$[P \Leftrightarrow Q] \Rightarrow [Q \Leftrightarrow P]$$

$$Q \Leftrightarrow P$$

Here is an example in which we make use of all three of our new subroutines.

Example: Show that $P \Leftrightarrow Q$, $P \Leftrightarrow [R \vee S]$, $S \vdash Q$.

	Demonstration Outline	Program Outline
1)	$P \Leftrightarrow Q$	Hyp
2)	$P \Leftrightarrow [R \vee S]$	Hyp
3)	$[R \vee S] \Leftrightarrow P$	CPB, For 2
4)	$[R \vee S] \Leftrightarrow Q$	TPB, For 3, For 1
5)	S	Hyp
6)	$S \Rightarrow [R \vee S]$	Taut
7)	$R \vee S$	MP, For 6, For 5
8)	Q	MPB, For 4, For 7

An examination of the demonstrations we have made thus far in this chapter will show that we quite often want to infer a formula $\mathfrak{G}$ that has been obtained from a formula $\mathfrak{F}$ by substituting a certain formula for one or more of the formulas that appear in $\mathfrak{F}$. For example, consider the demonstration outlined below.

	Demonstration Outline	Program Outline
1)	$P \wedge R$	Hyp
2)	$P \Leftrightarrow S$	Hyp
3)	P	RCS, For 1
4)	S	MPB, For 2, For 3
5)	R	LCS, For 1
6)	$S \wedge R$	CI, For 4, For 5

This shows that $P \wedge R,\ P \Leftrightarrow S \vdash S \wedge R$, or in other words, under the given hypotheses, "S" may be substituted for "P" in " $P \wedge R$ ". We will now introduce a new subroutine that goes right to the heart of the matter and simplifies such proofs. This subroutine is based on a principle which generalizes the Substitution Principle for tautologies which we introduced in Section 1.8. That principle says that

> If $\mathfrak{S}$ and $\mathfrak{R}$ are equivalent formulas (i.e., if $\mathfrak{S} \Leftrightarrow \mathfrak{R}$ is a tautology) and if $\mathfrak{G}$ is a formula obtained from the formula $\mathfrak{F}$ by substituting $\mathfrak{R}$ for $\mathfrak{S}$ in zero, one, or more occurrences of $\mathfrak{S}$ in $\mathfrak{F}$, then $\mathfrak{G}$ and $\mathfrak{F}$ are equivalent (i.e., $\mathfrak{G} \Leftrightarrow \mathfrak{F}$ is a tautology).

In the generalized principle, it is not required that $\mathfrak{S}$ and $\mathfrak{R}$ be equivalent.

General Substitution Principle: If $\mathfrak{F}$, $\mathfrak{G}$, $\mathfrak{R}$, and $\mathfrak{S}$ are well formed formulas such that $\mathfrak{G}$ is obtained from $\mathfrak{F}$ by substituting $\mathfrak{R}$ for $\mathfrak{S}$ in zero, one, or more occurrences of $\mathfrak{S}$ in $\mathfrak{F}$, then $[\mathfrak{S} \Leftrightarrow \mathfrak{R}] \Rightarrow [\mathfrak{G} \Leftrightarrow \mathfrak{F}]$ is a tautology.

Referring to truth table ($\mathfrak{J}$), we see that the task of showing that this formula is a tautology reduces to that of showing that $\mathfrak{G} \Leftrightarrow \mathfrak{F}$ has truth value t whenever $\mathfrak{S} \Leftrightarrow \mathfrak{R}$ does. When does $\mathfrak{S} \Leftrightarrow \mathfrak{R}$ have truth value t? Exactly when $\mathfrak{S}$ and $\mathfrak{R}$ have the same truth value.

So in order to justify the General Substitution Principle, all we have to do is show that $\mathfrak{F}$ and $\mathfrak{G}$ have the same truth value as each other whenever $\mathfrak{R}$ and $\mathfrak{S}$ have the same truth value as each other. But this is easy to see. Suppose that truth values have been assigned to the atomic formulas occurring in $\mathfrak{R}$, $\mathfrak{S}$, and $\mathfrak{F}$ in such a way that $\mathfrak{R}$ and $\mathfrak{S}$ have the same truth value. Let us now obtain $\mathfrak{G}$ from $\mathfrak{F}$ by replacing a certain number of occurrences of $\mathfrak{S}$ in $\mathfrak{F}$ by $\mathfrak{R}$. It is not difficult to see that although $\mathfrak{G}$ may be a different formula from $\mathfrak{F}$, from the truth value point of view nothing has changed. Since $\mathfrak{R}$ and $\mathfrak{S}$ have the same truth value, each replacement of an occurrence of $\mathfrak{S}$ in $\mathfrak{F}$ by $\mathfrak{R}$ made no change in the truth values of the component formulas, and so $\mathfrak{G}$ has the same truth value as $\mathfrak{F}$.

Example: Let us take " $[P \vee Q] \Rightarrow [P \wedge Q]$ " as the formula $\mathfrak{F}$, "P" as $\mathfrak{S}$, " $\sim Q$ " as $\mathfrak{R}$, and " $[P \vee Q] \Rightarrow [\sim Q \wedge Q]$ " as $\mathfrak{G}$. Check that these satisfy the conditions of the General Substitution Principle.

We now show in two ways that

$$[P \Leftrightarrow \sim Q] \Rightarrow [[[P \vee Q] \Rightarrow [\sim Q \wedge Q]] \Leftrightarrow [[P \vee Q] \Rightarrow [P \wedge Q]]]$$

is a tautology. First, directly:

P	Q	$\sim Q$	$P \wedge Q$	$P \vee Q$	$\sim Q \wedge Q$	$\mathfrak{F}$	$\mathfrak{G}$	$\mathfrak{G} \Leftrightarrow \mathfrak{F}$	$P \Leftrightarrow \sim Q$	$[P \Leftrightarrow \sim Q] \Rightarrow [\mathfrak{G} \Leftrightarrow \mathfrak{F}]$
t	t	f	t	t	f	t	f	f	f	t
t	f	t	f	t	f	f	f	t	t	t
f	t	f	f	t	f	f	f	t	t	t
f	f	t	f	f	f	t	t	t	f	t

Since all four entries in the last column are "t" the given formula is a tautology. Secondly, we use the method described in the discussion that follows the statement of the General Substitution Principle. According to this method we have to show that $\mathfrak{F}$ and $\mathfrak{G}$ have the same truth value whenever "P" and " $\sim Q$ " have the same truth value. Using the truth table above, this is easy. We see that "P" and " $\sim Q$ " have the same truth value in lines 2 and 3 of the table. Looking along these lines, we see that in each of them $\mathfrak{F}$ and $\mathfrak{G}$ have the same truth value as each other.

Let us now describe a useful subroutine based on the General Substitution Principle. If $\mathfrak{R}$, $\mathfrak{S}$, $\mathfrak{T}$, and $\mathfrak{G}$ are as described in the statement of that principle, then the following sequence of formulas would be useful during the course of many demonstrations:

1)	$\mathfrak{T}$	obtained previously in the demonstration
2)	$\mathfrak{S} \Leftrightarrow \mathfrak{R}$	
3)	$[\mathfrak{S} \Leftrightarrow \mathfrak{R}] \Rightarrow [\mathfrak{G} \Leftrightarrow \mathfrak{T}]$	Taut (guaranteed by the General Substitution Principle)
4)	$\mathfrak{G} \Leftrightarrow \mathfrak{T}$	MP, For 3, For 2
5)	$\mathfrak{G}$	MPB, For 4, For 1

So we build into our machine a subroutine which is such that, upon receiving the command "GSP, For 2, For 1", it prints formulas (3), (4), (5) above. As a very simple example, we could take the one with which we began this discussion: $P \wedge R,\ P \Leftrightarrow S \vdash S \wedge R$.

	Demonstration Outline	Program Outline
1)	$P \wedge R$	Hyp
2)	$P \Leftrightarrow S$	Hyp
3)	$S \wedge R$	GSP, For 2, For 1

Notice that, once again, the order in which the formulas are mentioned in this command is important.

You will undoubtedly have noticed that this substitution subroutine would have been most useful in those places where you have needed to substitute $\mathfrak{S}$ for $\sim[\sim\mathfrak{S}]$ (where $\mathfrak{S}$ is some wff). The need to do this has lengthened your demonstrations on many occasions. Now that we have the "GSP-subroutine" we can redesign our machine so that this problem is taken care of automatically. More specifically, we will arrange things so that henceforth whenever the machine carries out an order, before it prints the resulting formula it scans that formula looking for double negations. If it finds any, without your having to command it to do so, it immediately goes into the "GSP-subroutine" and prints a formula equivalent to the original one. This will simplify some of our work in outlining demonstrations.

Example: Show that $P,\ Q \Rightarrow {}^{\sim}P \vdash {}^{\sim}Q$.

	Demonstration Outline	Program Outline
1)	P	Hyp
2)	$Q \Rightarrow {}^{\sim}P$	Hyp
3)	$P \Rightarrow {}^{\sim}Q$	CPI, For 2
4)	${}^{\sim}Q$	MP, For 3, For 1

To give you an idea of the amount of work we are being saved, we reproduce below the machine printout for the demonstration outlined so simply above. (Actually, this is even more impressive if you remember that all the formulas involving "$\Leftrightarrow$" are abbreviations for something even more horrendous!)

1) P

2) $Q \Rightarrow {}^{\sim}P$

$${}^{\sim}[{}^{\sim}P] \Rightarrow {}^{\sim}Q$$

$${}^{\sim}[{}^{\sim}P] \Leftrightarrow P$$

$$[{}^{\sim}[{}^{\sim}P] \Leftrightarrow P] \Rightarrow [[P \Rightarrow {}^{\sim}Q] \Leftrightarrow [{}^{\sim}[{}^{\sim}P] \Rightarrow {}^{\sim}Q]]$$

$$[P \Rightarrow {}^{\sim}Q] \Leftrightarrow [{}^{\sim}[{}^{\sim}P] \Rightarrow {}^{\sim}Q]$$

3) $P \Rightarrow {}^{\sim}Q$

4) ${}^{\sim}Q$

One final remark before we proceed to some problems involving all these new subroutines. If, for some reason, you want the machine to print a formula with the double negations left in, you have only to order the machine to omit its automatic ${}^{\sim}$-elimination on that step of the program. This is done by adding ": Stop" to the command that is to produce that step. For example, in the demonstration outlined above, if you changed the command that results in the printing of formula (3) to "CPI, For 2: Stop" then, instead of formula (3), the machine would print

$${}^{\sim}[{}^{\sim}P] \Rightarrow {}^{\sim}Q$$

and then go on to try to execute the next command (in this particular case as it now stands, it would not succeed, of course).

PROBLEM SET 10

In Problems 1 - 8, outline a demonstration of the given result and provide a program outline for that demonstration.

1. Show that $P \Leftrightarrow S,\ Q \Leftrightarrow T \vdash [P \wedge Q] \Leftrightarrow [S \wedge T]$. 1 taut
2. Show that $P \Leftrightarrow [R \vee S],\ [R \vee S] \Rightarrow Q,\ P \vdash Q$. no taut
3. Show that $[{\sim}P \wedge S] \Leftrightarrow R,\ {\sim}P \Leftrightarrow S,\ {\sim}P \vdash {\sim}[P \vee {\sim}R]$. 1 taut - 9
4. Show that ${\sim}P \Leftrightarrow Q,\ Q \Leftrightarrow R,\ R \vdash {\sim}P$. none

Every taut has a Biconditional.

5. Show that $Q \Leftrightarrow S,\ R \Leftrightarrow {\sim}S,\ P \Leftrightarrow R \vdash P \Leftrightarrow {\sim}Q$. 1
6. Show that $Q \Leftrightarrow R,\ {\sim}P \Leftrightarrow Q,\ {\sim}R \vdash P$. 1
7. Show that $P \Leftrightarrow {\sim}S,\ Q \Leftrightarrow S \vdash P \Rightarrow {\sim}Q$. 1
8. Show that ${\sim}P,\ [R \vee Q] \Leftrightarrow P,\ S \Rightarrow [Q \vee R] \vdash {\sim}[P \vee S]$. ? tauts - 11 steps
9. Show that $R \Leftrightarrow P,\ {\sim}R \Rightarrow [{\sim}S \vee T],\ P \Leftrightarrow {\sim}[S \wedge {\sim}T] \vdash S \Rightarrow T$. 3 tauts - 12 lines
10. Show that $Q \Leftrightarrow [{\sim}R \wedge S],\ [S \vee {\sim}T] \Leftrightarrow R,\ Q \vdash T$. 3 - 11