CHAPTER 5
TRUTH TABLES

LOCATING AND NUMBERING CONNECTIVES

In this section we will learn how to number connectives. Perhaps we should begin with a complicated formula like the following one: -(--(P.-Q)v(R=(T>--S))). There are ten connectives in this formula. When we are finished we will have a number from 1 to 10 listed above each of these connectives.

There are several different schemes for numbering connectives. The one we prefer proceeds according to the following three simple rules:

1. We always start with those connectives that are inside the innermost set of parentheses, and proceed from inside out.
2. We always do tildes first.
3. We work from the right to the left.

Of these rules, the first is the most important. To follow it, all we need to know is how many pairs of left and right parentheses surround each connective. In the case we are examining, the tilde on the far left of the formula is inside no pairs of parentheses. The two tildes to its right are in one pair of parentheses, namely, the left and right ones at the far left and right ends of the formula. The next two connectives over -- the dot and the tilde -- are inside two pairs of parentheses. On the other hand, the wedge is inside only one pair of parentheses. While the "=" is in two pairs of parentheses, and the three remaining connectives are all located inside three pairs of parentheses.

Our first rule tells us that we should start with the three connectives located on the far right end of the formula. These are, the symbol we use to represent the horseshoe, ">," and the two tildes. After we have located these connectives our second rule comes into play. This rule, recall, tells us to do the tildes before doing the horseshoe, or any other binary connective. Which tilde should we do first, however? It is at this point that we use our third rule. It tells us that we should start at the right end of the formula and work left. So we should identify the tilde to the direct left of S "1," and we should label "2" the tilde to its immediate left.

```
                              2 1
- ( - - (P . - Q) v (R = (T > - - S)))
```

The only connective we have not yet done in three pairs of parentheses is the >. So we will do it next, and we will label it "3." We then get:

```
                            3 2 1
- ( - - (P . - Q) v (R = (T > - - S)))
```

Now that we've completed all those connectives embedded in three sets of parentheses, we turn next to the ones embedded in two pairs of parentheses. These are, the dot, the tilde next to Q, and the =. The second rule tells us to do tildes first, so the tilde to the direct left of Q must be 4. With respect to the dot and the =, since our third rule instructs us to work from right to left, 5 is =, and 6 is the dot.

```
           6 4         5    3 2 1
- ( - - (P . - Q) v (R = (T > - - S)))
```

We turn next to those connectives in only one set of parentheses. These include the two tildes to the left of (P.-Q) and the wedge in the middle of the formula. Here our second rule, however, tells us to do tildes before doing other connectives. Therefore, the two tildes in front of (P.-Q) need to be done before doing the wedge. Since our third rule tells us to work from right to left the tilde to the direct left of (P.-Q) should be 7. The tilde to its left will be 8, and the wedge will be 9.

87   64   9   5   3 21
- (- - (P . - Q) v (R = (T > - - S)))

The only connective we haven't numbered yet is the tilde on the far-left end of the formula. All we have to do is identify it as 10, and we are finished.

10 87   64   9   5   3 21
- (- - (P . - Q) v (R = (T > - - S)))

The last connective we do, i.e., the one with the highest number, we call "the main connective." It is the most important connective in the entire formula.

As you have probably already noticed, what we have been doing is really not any different from what is done in mathematics. For example, if we want to find the value of -(2+3)*(4/2), we first get the value of 2+3 (i.e., 5), and 4/2 (i.e., 2). Then we multiply negative 5 by 2. We have been going through the same procedure here. So numbering the connectives should be easy from now on.

You might also note that what we are doing here makes good sense in English. Clearly there is a big difference in meaning between the following claims:

Either both Albert and Barbara are happy, or Charles is happy.

Albert is happy, and besides that, either Barbara or Charles is happy.

To decide whether the first claim is true, we must first obtain the values of the claims, Albert and Barbara are happy, and Charles is happy. To decide whether the second claim is true, we first need to decide whether the claims that Albert is happy, and that Barbara or Charles is happy, are true. Notice also, that we represent these two claims in symbols differently. The first claim we represent as, ((A.B)vC), and we should view it as an or-claim. On the other hand, the second claim is expressed in symbols as, (A.(BvC)), and it is an and-claim.

A PROBLEM

Instructions: Number the connectives in the formula below.

(- (- P. (Q v - R)) = (S > T))

SETTING UP THE TABLE

In setting up a truth table, the first thing we need to do is to find out how many different letters occur in the formula. For example in the formula, ((P.-Q)>-(-R=-(P.R))), only three different letters occur, namely P, Q, and R. We list these letters in alphabetical order from left to right.

Immediately after we have done this we list the formula we want to test. If that formula is a single statement we list it to the direct right of the letters. If, instead, it is a set of statements, we replace the first semicolon with a dot and surround the first and second set members with a pair of parentheses. Once we have done this, we then take this unit and replace the semicolon to its immediate right (if there is one) with a dot. Then, we surround it and the next set member with a pair of parentheses. We continue this process until we have conjoined all the set members with dots and surrounded them with parentheses. The last step is to delete the curly braces at the beginning and end of the formula. This formula we then put directly to the right of the letters we have listed. Suppose, for example, we want to test the following set of statements: {(P>-Q); (-R=T); --S}. The process outlined suggests that we go through the following steps:

1. {(P > - Q) . (- R = T); - - S}
2. {((P > - Q) . (- R = T)); - - S}
3. {((P > - Q) . (- R = T)) . - - S}
4. {(((P > - Q) . (- R = T)) . - -S)}
5. (((P > - Q) . (- R= T)) . - - S)

The formula we have obtained by going through this process will be a single statement. It will be placed to the right of the letters in our table.

We treat arguments similarly to sets of statements. Thus, we convert the semicolon that separates the first two premises into a dot, and we surround these two premises by a pair of parentheses. We continue this process until all the premises have parentheses around them, and they are all conjoined with dots. Finally, we remove the curly braces at the beginning and end of the premises. Once we have finished this, we replace the backslash with a >, and we then surround the entire formula with a pair of parentheses. This is the statement we will place to the right of the letters and test.

Suppose, for example, we want to test the following argument: {(P>(Q.R)); (Q=-T); --T} / (-R>-P). We do the following:

1. {(P > (Q . R)) . (Q = - T); - - T} / (- R > - P)
2. {((P > (Q . R)) . (Q = - T)); - - T} / (- R > - P)
3. {((P > (Q . R)) . (Q = - T)) . - - T} / (- R > - P)
4. {(((P > (Q . R)) . (Q = - T)) . - - T)} / (- R > - P)
5. (((P > (Q . R)) . (Q = - T)) . - - T) / (- R > - P)
6. (((P > (Q . R)) . (Q = - T)) . - - T) > (- R > - P)
7. ((((P > (Q . R)) . (Q = - T)) . - - T) > (- R > - P))

Why are we doing this? This question must wait for an answer until later. For the moment, we need only know that the statement we are going to test tells us something important about the set of statements or argument we are examining.

Once we have listed the different letters and the formula, our next task is to decide how many rows in the table we need to build. This will depend entirely on how many different letters we have listed. To find out how many rows to build, all we need to do is use the formula 2^n, where n is the number of different letters. Thus, if three letters occur in the statement, the formula tells us to build 2^3 (= 8) rows. If it contains four letters, the formula tells us to build 2^4 (= 16) rows, etc. (Note: The number of rows doubles every time.)

We build the rows in the following way. In the rows directly under the leftmost letter, we begin by listing T's, and continue listing them until we have filled half the rows. We then switch to F's, and fill the remaining rows. We then turn to the column under the next letter to the right. In the rows under it we list half as many T's as we did before, followed by half as many F's. This process, we then repeat until we complete all of the rows.

Let's see a practical example of this. Suppose the formula we want to test is, ((R v - G) > - (- S . R)). The table should be set up as follows:

GRS	((Rv - G)>(- S . R))
TTT	
TTF	
TFT	
TFF	
FTT	
FTF	
FFT	
FFF	

What have we done here, and why have we done it? Each row in our table represents one way the world might be. In the first row, for example, where P, Q, and R are all true, we are representing the possibility that all three of our atomic statements are true. Suppose the atomic statement P represents the claim that Paul is happy, while Q stands for Quincy is happy, and R is Reginald is happy. This row represents the possibility that all three of these individuals are happy. The last row, on the other hand, represents the possibility that none of them are happy. By building the table in the way we have, we will have shown every possible way the world might be with respect to these three individual statements.

QUESTIONS:

Consider the following set of statements:

{- - P; (L > - (T . U)); (L v - T)}

1. How many different letters are contained in this set?
2. List those letters in alphabetical order, and translate it into a single statement.
3. How many rows will we have to build in this table?
4. Beneath the letter L how many rows of T's will occur before an F occurs?
5. But what are we to do about P? How many rows under it will contain T's before we find an F?

CONSTRUCTING TRUTH TABLES

By now you probably can assign numbers to the connectives and set the truth table up. All you need to learn is how to fill the values for the various connectives in, for each row of the table, and how to interpret the result. In this section, we will try to teach you how to fill the values in for each connective.

Once you have the basics down, doing more sophisticated problems should be easy. So let's begin with a simple case.

```
    76  432    5 1
PQ - - ( - - - P . - Q)
TT
TF
FT
FF
```

We have set the table up in the manner suggested earlier. Unfortunately, the hardest part of constructing it still lies ahead of us. Before we are finished we need to have a set of values listed in each row of the table under every connective.

We begin with the first column. The connective in this column is a tilde. Tildes flip values. Moreover, it's clear that since this one is to the immediate left of Q it flips the value of Q. As we can see, however, in the first row Q has the value of T. So we assign the value F to -Q.

```
    76  432    5 1
PQ - - ( - - - P . - Q)
TT                 F
TF                 T
FT                 F
FF                 T
```

In the second row Q has the value F, so -Q has the value T. We therefore place a T in column 1, row 2. What value should we place under column 1, in row 3, and what about the last row in column 1? What is its value? If you answered "F" to the first question and "T" to the second you are right.

Now it's your turn to complete a column. What values should we list in rows 1-4 of column 2?
If you answered "FFTT," you got it, and the table will then look like this:

```
    76  432    5 1
PQ - - ( - - - P . - Q)
TT           F     F
TF           F     T
FT           T     F
FF           T     T
```

Now let's turn to column 3. It is also a tilde. Moreover, it is to the direct left of column 2. It therefore flips the values we just got under column 2. Therefore, we get the same values in column 3 that we had under P when we started. A double negative is, as you might expect, the same thing as the affirmative.

```
     76  432    5 1
PQ - - (- - - P . - Q)
TT        TF     F
TF        TF     T
FT        FT     F
FF        FT     T
```

How would you complete the values for column 4? You should have answered "FFTT," and your table will then look like this:

```
     76  432    5 1
PQ - - (- - - P . - Q)
TT       FTF     F
TF       FTF     T
FT       TFT     F
FF       TFT     T
```

Next we come to column 5. It's a dot, not a tilde. Dots differ significantly from tildes. They are binary connectives, and so, unlike tildes that have only a right side, they have both left and right sides. Also, unlike tildes, dots always carry a pair of parentheses with them. In the formula we are considering, the dot causes a pair of parentheses.

Now dot claims are true only when both sides of those claims are true. (Just as in English, "Albert is happy and Barbara is happy," is true only if both the claims that Albert is happy and that Barbara is happy are true.)

What, however, is the claim on the left side of the dot, and what is the claim on the right side of the dot? Clearly the claim on the left side is ---P, while -Q is the claim on its right side. So the values on both these sides are the values we must use in determining whether the dot claim is true or false. In both cases, the highest numbered connective provides the values for that formula. To find the value for the left side of the dot just locate the highest number left of the dot until you come to the left parenthesis (viz., 4). To find the value for the right side of the dot locate the highest number on the right of the dot until you hit the right parenthesis (viz., 1). The dot claim will be true, in a row, only when both these sides are true. If you look at the values of the two sides in row 1 you will see that they are both false in this row. Therefore, the value of the dot is F in this row.

In row 2, the value of the left side of the dot is again F, but this time, the value of its right side is T. Still, this is not a case where both of the sides are T. So the dot is F in row 2. In row 3, while the claim on the left side of the dot is T, its right side is F. Here too then, the value of the dot is F. What about the last row, however? Which value does it have? I'll give you a hint: It's either a T or F.

Do you understand this? You should have answered "T," and the table should now look like this:

```
     76  432    5 1
PQ - - (- - - P . - Q)
TT       FTF  FF
TF       FTF  FT
FT       TFT  FF
FF       TFT  TT
```

All that remains to be done are the tildes at the left end of the formula. Tildes always flip values. It should be clear that 7 flips the value of 6, but what value does 6 flip? When tildes occur to the direct left of a left parenthesis they flip the highest numbered connective within that set of parentheses. Since the dot is the highest numbered connective within the parentheses, column 6 flips the value of column 5 in each row. Consequently, the completed table should look like this:

```
    76  432    5 1
PQ - - ( - - - P . - Q)
TT FT FTF  FF
TF  FTFTF  FT
FT  FTTFT  FF
FF  TFTFT  TT
```

We still need to consider the other connectives. First, however, let's try one more problem.

```
     5  1   2     4 3
PQ - ( - P . Q) .  - Q)
TT
TF
FT
FF
```

The connective under column 1 is a tilde, and tildes flip values. So the value listed under column 1, in each row, should be the opposite of the value of P in that row. Therefore, column 1's values should read:

```
     5  1   2     4 3
PQ - ( - P . Q) .  - Q)
TT     F
TF     F
FT     T
FF     T
```

The connective in column 2 is a dot. The formula on the left side of the dot is -P, and column 1 contains its values. Meanwhile, the formula on the right side of the dot is Q, and the column under the initial Q lists these values. Since dot claims are true only when both sides of the dot are true, the only row in which both sides of the dot are true is row 3. So, the values listed in column 2 should be:

```
     5  1   2     4 3
PQ - ( - P . Q) .  - Q)
TT     F   F
TF     F   F
FT     T   T
FF     T   F
```

We now need to figure out the values in column 3. Like column 1, however, column 3 is a tilde, and tildes flip values. Obviously, then, in each row of column 3 the values listed should be the opposite of the values listed for Q in that row. Thus, we get:

```
     5  1   2     4 3
PQ - ( - P . Q) .  - Q)
TT     F   F       F
TF     F   F       T
FT     T   T       F
FF     T   F       T
```

Next, we need to do column 4. It's a dot, and dots are binary connectives. So the dot must have a left and a right side. The formula on the left side of the dot is (-P.Q) and the values of this formula are listed under the highest numbered connective in that formula, namely, column 2. The formula on the right side of the dot is -Q and the values of this formula are listed under column 3. Now we know that dotted claims are true only when both of their sides are true. So, column 4 will be true only when both columns 2 and 3 are true. However, there are no rows in which columns 2 and 3 are both true. Therefore, all of the values in column 4 should be false. Once we fill those values in our table will look like this:

	5	1	2	4	3
PQ	- (	- P	. Q)	.	- Q)
TT		F	F	F	F
TF		F	F	F	T
FT		T	T	F	F
FF		T	F	F	T

Column 5 is a tilde, and tildes flip values. However, what column's values is column 5 supposed to flip? It flips the values of the highest numbered connective in the parentheses, namely 4. Thus, the completed table should read:

	5	1	2	4	3
PQ	- (	- P	. Q)	.	- Q)
TT	T	F	F	F	F
TF	T	F	F	F	T
FT	T	T	T	F	F
FF	T	T	F	F	T

The next connective is the wedge. Like the dot, the wedge is a binary connective, and so, it has a left and a right side. Wedge claims are false, however, only when both sides are false. In all other cases the wedge claim is true.

We'll move on soon. First, however, let's look at one quick problem that involves this connective.

	1	3	2
AN	((A . N)	v	- N)
TT			
TF			
FT			
FF			

Column 1 and 2 should be easy. The connective in column 1 is a dot, and it glues the atomic letters A and N together. Since dots are true only when both sides are true, we should get the following values under column 1:

	1	3	2
AN	((A . N)	v	- N)
TT	T		
TF	F		
FT	F		
FF	F		

We turn now to column 2, which is obviously a tilde. Since tildes flip values, the values in column 2 should read:

	1	3	2
AN	((A . N)	v	- N)
TT	T		F
TF	F		T
FT	F		F
FF	F		T

We can now do column 3. It's a wedge, and wedges are false only when both of their sides are false. Since the only row in which this happens is row 3, however, the values under column 3 should read:

	1	3	2
AN	((A . N)	v	- N)
TT	T	T	F
TF	F	T	T
FT	F	F	F
FF	F	T	T

The next connective is the =. Like the dot and wedge, = is a binary connective and so has both a left and right side. It is true, however, only when both of its sides have the same value. In those cases where its two sides have different values the = gets the value F.

This connective is easy. So let's try a quick example here.

	2	3	1
KL	((L v K)	=	(K = L))
TT			
TF			
FT			
FF			

Clearly we need to do column 1 first. The values in it should be:

	2	3	1
KL	((L v K)	=	(K = L))
TT			T
TF			F
FT			F
FF			T

Next we do column 2. When we have completed it our table will look like this:

	2	3	1
KL	((L v K)	=	(K = L))
TT	T		T
TF	T		F
FT	T		F
FF	F		T

Finally, we can now figure out the values in column 3. Our completed table will then look like this:

	2	3	1
KL	((L v K)	=	(K = L))
TT	T	T	T
TF	T	F	F
FT	T	F	F
FF	F	F	T

Now for the last, but trickiest, of the connectives: Like . , v, and =, > is a binary connective and so has a left and a right side. It is only false in one case, however, namely when its left side is T and its right side is F. It is T in all other cases. (Check your text, or ask your teacher if you want to know why we evaluate it in this way.)

Let's look at an example that uses this connective. Then you will have all of the fundamentals for constructing tables.

	2	4	3	1
PQ	((Q > P)	>	-	(P > Q))
TT				
TF				
FT				
FF				

Here we begin with column 1. Remember it is false only when the formula on its left side is true and the formula on its right side is false. Reading down the column, what values do we get?

	2	4	3	1
PQ	((Q > P)	>	-	(P > Q))
TT				T
TF				F
FT				T
FF				T

Now do column 2. Here we must be careful, however. Q occurs on the left side of the > and P on the right. So (Q>P) is false only when Q is true and P false, and that occurs only in the third row.

	2	4	3	1
PQ	((Q > P)	>	-	(P > Q))
TT	T			T
TF	T			F
FT	F			T
FF	T			T

We must now do column 3, and its values obviously reverse the values in column 1. So we should get:

	2	4	3	1
PQ	((Q > P)	>	-	(P > Q))
TT	T		F	T
TF	T		T	F
FT	F		F	T
FF	T		F	T

Finally, we can do column 4. It uses the values of column 2 on the left and column 3 on the right. Once we have done this column, the completed table should look like this:

	2	4	3	1
PQ	((Q > P)	>	-	(P > Q))
TT	T	F	F	T
TF	T	T	T	F
FT	F	T	F	T
FF	T	F	F	T

As far as the fundamentals of constructing truth tables are concerned that is it. The only thing you may not know yet is how to interpret them.

EVALUATING TRUTH TABLES

1. EVALUATING SINGLE STATEMENTS:

Once you have completed the table evaluating it is easy. If the formula is a single statement it will be logically true if it contains only T's in its highest numbered column. While if it has nothing but F's under this connective, it will be logically false. Finally, if there is a mixture of T's and F's, it will be logically indeterminate.

The reason for this is simple enough to understand. Each row in the table represents one way the world might be. The value under the highest numbered connective tells us whether the formula is true or false in that possible world. However, in our table we have considered all the ways the world could be. If the formula is true in all these cases then it must be true. On the other hand, if it is false in all these cases then it must be false. While if there is a mixture of T and F's the statement could be true and it could be false. Logic alone cannot tell which it is.

EXAMPLES

PQ	(P > (Q > P))	((P . Q) . - Q)	(P = - Q)
TT	T T	T FF	FF
TF	T T	F FT	TT
FT	T F	F FF	TF
FF	T T	F FT	FT
	↑	↑	↑
	LOGICALLY TRUE	LOGICALLY FALSE	LOGICALLY INDETERMINATE

2. EVALUATING SETS OF STATEMENTS:

If the formula contains curly braces it will be either a set of statements or an argument. If it contains a backslash it will be an argument, while if it does not it will be a set of statements.

Sets of statements are either consistent or inconsistent. A set of statements will be consistent if at least one T occurs under the formula's main connective (i.e., in its highest numbered column). On the other hand, if all the values under the highest numbered connective are F's, the set of statements will be inconsistent. The reason for this should be clear. When we say a set of statements is consistent, what this means is that all of the statements in that set could be true together. While when we say a set of statements is inconsistent, this means that it isn't possible for all of the statements in that set to be true together.

What we have done here is to convert the set of statements into a single statement. That single statement asserts that all of the members of our original set of statements are true. Clearly, however, if this single statement even can be true this will show that our set of statements is consistent; while if it cannot be true, this will show that our set is inconsistent.

EXAMPLE 1

This is a set. {(P > Q); - (Q > P)}

This is a single statement that asserts that all the members in the set are true.

PQ	((P > Q) . - (Q > P))
TT	T F F T
TF	F F F T
FT	T T T F
FF	T F F T
	↑

We find a T in row three of the indicated column. This shows that the statement that asserts that all of the set members are true can be a true statement, and so, it establishes that the set is consistent.

EXAMPLE 2

This is a set.

{(((P > Q); P; -Q}

This is a single statement that asserts that all the members in the set are true.

PQ (((P > Q) . P) . - Q)

PQ	(((P > Q)	. P)	.	- Q)
TT	T	T	F	F
TF	F	F	F	T
FT	T	F	F	F
FF	T	F	F	T
			↑	

Here we find an F in every row of the indicated column. This tells us the statement that asserts that all the members of the set are true cannot be true. It tells us, therefore, that the set is inconsistent.

3. EVALUATING ARGUMENTS:

When we say that an argument is valid what this means is that it is not possible for it to have all true premises and a false conclusion. The statement we have constructed, however, asserts that if all of the premises are true then the conclusion will be true. If this claim is always true (i.e., true in every row of the table), the argument cannot have all true premises and a false conclusion. It will, therefore, be valid. Conversely, if there is even one false row in our table this establishes that it's possible for the argument to have all true premises and a false conclusion. So, it tells us that the argument is invalid.

EXAMPLE 1

This is an argument.

{(((P > Q); - P} / - Q

This is a statement that asserts that if the argument's premises are true its conclusion will be true.

PQ (((P > Q) . - P) > - Q)

PQ	(((P > Q)	.	- P)	>	- Q)
TT	T	F	F	T	F
TF	F	F	F	T	T
FT	T	T	T	F	F
FF	T	T	T	T	T
				↑	

The value F occurs in the third row of the indicated column. So the claim that if the premises of the argument are true its conclusion will be true is sometimes false. Therefore it is possible for the argument to have all true premises and a false conclusion. So it is invalid.

EXAMPLE 2

This is an argument.

{(((P = - Q); Q} / - P

This is a statement that asserts that if the argument's premises are true its conclusion will be true.

PQ (((P = - Q) . Q) > - P)

PQ	(((P =	- Q)	. Q)	>	- P)
TT	F	F	F	T	F
TF	T	T	F	T	F
FT	T	F	T	T	T
FF	F	T	F	T	T
				↑	

All of the rows have the value T in the indicated column. So the claim that if the premises of the argument are true, its conclusion is true, has to be true. Therefore, the argument is valid.

4. EVALUATING PAIRS OF STATEMENTS:

You may recall that in chapter 1 we briefly introduced the idea of logical equivalence. We said two statements are logically equivalent just in case they must have the same truth-values. How can we use truth tables to decide if a pair of statements is logically equivalent? The procedure here is simple. Just connect the two statements with =, and surround them with a set of parentheses. Test this statement for logical truth. If it is logically true, the pair of statements in question is logically equivalent. On the other hand, if the single statement we have constructed is not logically true, the pair of statements is not logically equivalent.

Suppose, for example, that the two statements are (P= - Q), and - (P= Q). To test this pair of statements for logical equivalence, we connect them with =, and surround them with a pair of parentheses. Doing this, we obtain:

((P = - Q) = - (P = Q))

We then test this statement for logical truth.

PQ	(((P = - Q)		=	-	(P = Q)))
TT	F	F	T	F	T
TF	T	T	T	T	F
FT	T	F	T	T	F
FF	F	T	T	F	T

↑

This statement is logically true.
Therefore, the pair of statements
is logically equivalent.

PROBLEMS

A. SINGLE STATEMENTS

Instructions: Determine whether the following statements are logically true, logically false, or logically indeterminate by using the Truth Table Method. If you want to check your answer, go to the "Truth Table" chapter of <u>*Logical Reasoning*</u>*, enter the section of the chapter entitled "Original Problem," and type in the formula exactly as it appears below. The solution will appear on the screen.*

1. (((P>Q).(-P>-Q))=((P.Q)v(-P.-Q)))

2. ((P>(-R>-Q)).-(-(P.Q)vR))

3. (((PvQ)>(R.S))>((P>R).(P>S)))

4. (((P=-Q).(Q=-R))>(P.(QvR)))

5. ((Pv(Q.R))>-((PvQ).(-P>R)))

B. PAIRS OF STATEMENTS

Instructions: Determine whether the following pairs of statements are logically equivalent by using the Truth Table Method. To use this method, glue the two statements together with = and enclose the result with a pair of parentheses. Thus, the first problem below should be written: (-(-P.-Q)=(PvQ)). If the statement you have thus constructed is logically true, the pair of statements will be logically equivalent. (As before, you can check your results with the computer by going to the "Original Problem" section of the chapter on "Truth Tables and entering the formula.)

1. - (- P. -Q)	(P v Q)
2. (P > Q)	(Q > P)
3. (P. (Q = R))	(Q > (P . R))
4. (P> (Q > R))	((- P v - Q) v R)
5. ((P = Q) . (Q = R))	P = (Q = R))

C. SETS OF STATEMENTS

Instructions: Determine whether the sets of statements below are consistent or inconsistent by using the Truth Table Method. As before, you can use the "Original Problem" section of the chapter on Truth Tables to check your results.

Note: In the "Evaluating Truth Tables" portion of the Truth Table Tutorial you were told how to use the Truth Table Method to determine whether a set of statements is consistent or inconsistent. If you are using the "Original Problem" section of the program, you should simply type in the whole set of statements exactly as it appears below. The program will convert this set to the appropriate single statement and provide you with the result.

1. {(P > - Q); (Q > - P); (- P = Q)}

2. {(P > (Q . - R)); (- (P > Q) v -(P > - R))}

3. {(P > Q); (- R > - Q); (- R v P); - (P = R)}

4. {((P v Q) > R); (R =-S); (S > (- P v - Q))}

5. {- ((P v Q) v R); ((R . S) v (R . - S))}

D. ARGUMENTS

Instructions: Determine whether the arguments below are valid or invalid by using the Truth Table Method. Use the "Original Problem" section of the chapter on Truth Tables to check your results.

Note: In the "Evaluating Truth Tables" portion of the Truth Table Tutorial you were told how to use the Truth Table Method to determine whether an argument is valid or invalid. If you are using the "Original Problem" section of the program you should simply type the argument in exactly as it appears below. The program will convert this argument into the appropriate single statement and provide you with the result.

1. {((P . Q) > - R); (R > Q)} / - P

2. {- (P > (Q . R)); ((Q > - R) > - S)} / (P . - S)

3. {(P > (Q v R)); (Q > - S)} / (- P v (R . - S))

4. {((P . - Q) v (- P . Q)); (Q > - R); (- R > - Q)} / (P = R)

5. {((- P v Q) . (- Q = - R)); (P . - R)} / - S

E. INTRODUCING NEW CONNECTIVES

*Instructions: Suppose we introduce two new binary connectives. * is true only when its left side is true and its right side is false; while # is false only when both its left and right sides are true. Determine whether the* *following single statements are logically true, logically false, or logically indeterminate by using the Truth Table Method.*

1. ((P * Q) # (Q * P))

2. ((P # (Q * R)) * P)

3. ((P * (Q # R)) * P)

4. ((P > (Q * P)) # ((Q * P) = - P))

5. (((- P v Q) * - R) # (- P . - (R # Q)))

BRAINTEASER

A MURDER ON CHICKEN RANCH

The clues are:

1. ((♣ ⛏ ♥) = (♥ ⛏ ♣)) is logically true.
2. (♣ ⛏ ♣) is logically false.
3. (♣ ⛏ ♥) is not equivalent to (♣ • - ♣).
4. (♣ ⊤ ♥) is equivalent to (- ♥ • - ♣).

The translation key is:

A: An axe was used
C: The cook did it because the menu called for chicken Kiev.
G: The gardener did it because all the seeds in the garden were gone.
K: A knife was used.
M: The maid did it because there were feathers all over the house.
P: Poison was used.

The known facts are:

1. ((M ⛏ G) ⛏ C)
2. - (- M ⛏ K)
3. ((K ⊤ G) v (C ⊤ - C))
4. ((K ⛏ P) • (A ⛏ M))

What happened?