The connectives $\land$, $\lor$, and $\neg$ are truth-functional connectives. Recall what this means: the truth value of a complex sentence built by means of one of these symbols can be determined simply by looking at the truth values of the sentence’s immediate constituents. So to know whether $P \lor Q$ is true, we need only know the truth values of $P$ and $Q$. This particularly simple behavior is what allows us to capture the meanings of truth-functional connectives using truth tables.

Other connectives we could study are not this simple. Consider, the sentence *it is necessarily the case that S*. Since some true claims are necessarily true, that is, could not have been false (for instance, $a = a$), while other true claims are not necessarily true (for instance, $\text{Cube}(a)$), we can’t figure out the truth value of the original sentence if we are only told the truth value of its constituent sentence $S$. *It is necessarily the case*, unlike *it is not the case*, is not truth-functional.

The fact that the Boolean connectives are truth functional makes it very easy to explain their meanings. It also provides us with a simple but powerful technique to study their logic. The technique is an extension of the truth tables used to present the meanings of the connectives. It turns out that we can often calculate the logical properties of complex sentences by constructing truth tables that display all possible assignments of truth values to the atomic constituents from which the sentences are built. The technique can, for example, tell us that a particular sentence $S$ is a logical consequence of some premises $P_1, \ldots, P_n$. And since logical consequence is one of our main concerns, the technique is an important one to learn.

In this chapter we will discuss what truth tables can tell us about three related logical notions: the notions of logical consequence, logical equivalence, and logical truth. Although we’ve already discussed logical consequence at some length, we’ll tackle these in reverse order, since the related truth table techniques are easier to understand in that order.
Section 4.1
Tautologies and logical truth

We said that a sentence $S$ is a logical consequence of a set of premises $P_1, \ldots, P_n$ if it is impossible for the premises all to be true while the conclusion $S$ is false. That is, the conclusion must be true if the premises are true.

Notice that according to this definition there are some sentences that are logical consequences of any set of premises, even the empty set. This will be true of any sentence whose truth is itself a logical necessity. For example, given our assumptions about FOL, the sentence $a = a$ is necessarily true. So of course, no matter what your initial premises may be, it will be impossible for those premises to be true and for $a = a$ to be false—simply because it is impossible for $a = a$ to be false! We will call such logically necessary sentences logical truths.

The intuitive notions of logical possibility and logical necessity have already come up several times in this book in characterizing valid arguments and the consequence relation. But this is the first time we have applied them to individual sentences. Intuitively, a sentence is logically possible if it could be (or could have been) true, at least on logical grounds. There might be some other reasons, say physical, why the statement could not be true, but there are no logical reasons preventing it. For example, it is not physically possible to go faster than the speed of light, though it is logically possible: they do it on Star Trek all the time. On the other hand, it is not even logically possible for an object not to be identical to itself. That would simply violate the meaning of identity. The way it is usually put is that a claim is logically possible if there is some logically possible circumstance (or situation or world) in which the claim is true. Similarly, a sentence is logically necessary if it is true in every logically possible circumstance.

These notions are quite important, but they are also annoyingly vague. As we proceed through this book, we will introduce several precise concepts that help us clarify these notions. The first of these precise concepts, which we introduce in this section, is the notion of a tautology.

How can a precise concept help clarify an imprecise, intuitive notion? Let’s think for a moment about the blocks language and the intuitive notion of logical possibility. Presumably, a sentence of the blocks language is logically possible if there could be a blocks world in which it is true. Clearly, if we can construct a world in Tarski’s World that makes it true, then this demonstrates that the sentence is indeed logically possible. On the other hand, there are logically possible sentences that can’t be made true in the worlds you can
build with Tarski’s World. For example, the sentence

\[ \neg (\text{Tet}(b) \lor \text{Cube}(b) \lor \text{Dodec}(b)) \]

is surely logically possible, say if \( b \) were a sphere or an icosahedron. You can’t build such a world with Tarski’s World, but that is not logic’s fault, just as it’s not logic’s fault that you can’t travel faster than the speed of light. Tarski’s World has its non-logical laws and constraints just like the physical world.

The Tarski’s World program gives rise to a precise notion of possibility for sentences in the blocks language. We could say that a sentence is \( \text{tw-possible} \) if it is true in some world that can be built using the program. Our observations in the preceding paragraph could then be rephrased by saying that every \( \text{tw-possible} \) sentence is logically possible, but that the reverse is not in general true. Some logically possible sentences are not \( \text{tw-possible} \).

It may seem surprising that we can make such definitive claims involving a vague notion like logical possibility. But really, it’s no more surprising than the fact that we can say with certainty that a particular apple is red, even though the boundaries of the color red are vague. There may be cases where it is hard to decide whether something is red, but this doesn’t mean there aren’t many perfectly clear-cut cases.

Tarski’s World gives us a precise method for showing that a sentence of the blocks language is logically possible, since whatever is possible in Tarski’s World is logically possible. In this section, we will introduce another precise method, one that can be used to show that a sentence built up using truth-functional connectives is logically necessary. The method uses truth tables to show that certain sentences cannot possibly be false, due simply to the meanings of the truth-functional connectives they contain. Like the method given to us by Tarski’s World, the truth table method works only in one direction: when it says that a sentence is logically necessary, then it definitely is. On the other hand, some sentences are logically necessary for reasons that the truth table method cannot detect.

Suppose we have a complex sentence \( S \) with \( n \) atomic sentences, \( A_1, \ldots, A_n \). To build a truth table for \( S \), one writes the atomic sentences \( A_1, \ldots, A_n \) across the top of the page, with the sentence \( S \) to their right. It is customary to draw a double line separating the atomic sentences from \( S \). Your truth table will have one row for every possible way of assigning TRUE and FALSE to the atomic sentences. Since there are two possible assignments to each atomic sentence, there will be \( 2^n \) rows. Thus if \( n = 1 \) there will be two rows, if \( n = 2 \) there will be four rows, if \( n = 3 \) there will be eight rows, if \( n = 4 \) there will be sixteen rows, and so forth. It is customary to make the leftmost column have the top half of the rows marked TRUE, the second half FALSE. The next
column splits each of these, marking the first and third quarters of the rows with TRUE, the second and fourth quarters with FALSE, and so on. This will result in the last column having TRUE and FALSE alternating down the column.

Let’s start by looking at a very simple example of a truth table, one for the sentence $\text{Cube}(a) \lor \neg \text{Cube}(a)$. Since this sentence is built up from one atomic sentence, our truth table will contain two rows, one for the case where $\text{Cube}(a)$ is true and one for when it is false.

<table>
<thead>
<tr>
<th>$\text{Cube}(a)$</th>
<th>$\text{Cube}(a) \lor \neg \text{Cube}(a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

In a truth table, the column or columns under the atomic sentences are called *reference columns*. Once the reference columns have been filled in, we are ready to fill in the remainder of the table. To do this, we construct columns of T’s and F’s beneath each connective of the target sentence $S$. These columns are filled in one by one, using the truth tables for the various connectives. We start by working on connectives that apply only to atomic sentences. Once this is done, we work on connectives that apply to sentences whose main connective has already had its column filled in. We continue this process until the main connective of $S$ has had its column filled in. This is the column that shows how the truth of $S$ depends on the truth of its atomic parts.

Our first step in filling in this truth table, then, is to calculate the truth values that should go in the column under the innermost connective, which in this case is the $\neg$. We do this by referring to the truth values in the reference column under $\text{Cube}(a)$, switching values in accord with the meaning of $\neg$.

<table>
<thead>
<tr>
<th>$\text{Cube}(a)$</th>
<th>$\text{Cube}(a) \lor \neg \text{Cube}(a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

Once this column is filled in, we can determine the truth values that should go under the $\lor$ by looking at the values under $\text{Cube}(a)$ and those under the negation sign, since these correspond to the values of the two disjuncts to which $\lor$ is applied. (Do you understand this?) Since there is at least one T in each row, the final column of the truth table looks like this.

<table>
<thead>
<tr>
<th>$\text{Cube}(a)$</th>
<th>$\text{Cube}(a) \lor \neg \text{Cube}(a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T F</td>
</tr>
<tr>
<td>F</td>
<td>T T</td>
</tr>
</tbody>
</table>
Not surprisingly, our table tells us that the sentence \( \text{Cube}(a) \lor \neg \text{Cube}(a) \) cannot be false. It is what we will call a \textit{tautology}, an especially simple kind of logical truth. We will give a precise definition of tautologies later. Our sentence is in fact an instance of a principle, \( P \lor \neg P \), that is known as the law of the excluded middle. Every instance of this principle is a tautology.

Let’s next look at a more complex truth table, one for a sentence built up from three atomic sentences.

\[(\text{Cube}(a) \land \text{Cube}(b)) \lor \neg \text{Cube}(c)\]

In order to make our table easier to read, we will abbreviate the atomic sentences by \( A \), \( B \), and \( C \). Since there are three atomic sentences, our table will have eight \( (2^3) \) rows. Look carefully at how we’ve arranged the T’s and F’s and convince yourself that every possible assignment is represented by one of the rows.

\[
\begin{array}{ccc|c}
A & B & C & (A \land B) \lor \neg C \\
T & T & T & T \\
T & T & F & F \\
T & F & T & T \\
T & F & F & F \\
F & T & T & T \\
F & T & F & F \\
F & F & T & T \\
F & F & F & F \\
\end{array}
\]

Since two of the connectives in the target sentence apply to atomic sentences whose values are specified in the reference column, we can fill in these columns using the truth tables for \( \land \) and \( \neg \) given earlier.

\[
\begin{array}{ccc|c|c}
A & B & C & (A \land B) & \lor \neg C \\
T & T & T & T & F \\
T & T & F & T & T \\
T & F & T & F & F \\
T & F & F & F & T \\
F & T & T & F & F \\
F & T & F & F & T \\
F & F & T & F & F \\
F & F & F & F & T \\
\end{array}
\]

This leaves only one connective, the main connective of the sentence. We fill in the column under it by referring to the two columns just completed, using the truth table for \( \lor \).
When we inspect the final column of this table, the one beneath the connective $\lor$, we see that the sentence will be false in any circumstance where $\text{Cube}(c)$ is true and one of $\text{Cube}(a)$ or $\text{Cube}(b)$ is false. This table shows that our sentence is not a tautology. Furthermore, since there clearly are blocks worlds in which $c$ is a cube and either $a$ or $b$ is not, the claim made by our original sentence is not logically necessary.

Let’s look at one more example, this time for a sentence of the form

$$\neg(A \land (\neg A \lor (B \land C))) \lor B$$

This sentence, though it has the same number of atomic constituents, is considerably more complex than our previous example. We begin the truth table by filling in the columns under the two connectives that apply directly to atomic sentences.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>$A \land (\neg A \lor (B \land C))$</th>
<th>$\neg(A \land (\neg A \lor (B \land C))) \lor B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

We can now fill in the column under the $\lor$ that connects $\neg A$ and $B \land C$ by referring to the columns just filled in. This column will have an $F$ in it if and only if both of the constituents are false.
We now fill in the column under the remaining $\land$. To do this, we need to refer to the reference column under $A$, and to the just completed column. The best way to do this is to run two fingers down the relevant columns and enter a $T$ in only those rows where both your fingers are pointing to $T$’s.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$\neg(A \land (\neg A \lor (B \land C))) \lor B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
</tbody>
</table>

We can now fill in the column for the remaining $\neg$ by referring to the previously completed column. The $\neg$ simply reverses $T$’s and $F$’s.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$\neg(A \land (\neg A \lor (B \land C))) \lor B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
</tbody>
</table>

Finally, we can fill in the column under the main connective of our sentence. We do this with the two-finger method: running our fingers down the reference column for $B$ and the just completed column, entering $T$ whenever at least one finger points to a $T$. 

Section 4.1
We will say that a tautology is any sentence whose truth table has only T’s in the column under its main connective. Thus, we see from the final column of the above table that any sentence of the form
\[
\neg(A \land (\neg A \lor (B \land C))) \lor B
\]
is a tautology.

You try it

1. Open the program Boole from the software that came with the book. We will use Boole to reconstruct the truth table just discussed. The first thing to do is enter the sentence \(\neg(A \land (\neg A \lor (B \land C))) \lor B\) at the top, right of the table. To do this, use the toolbar to enter the logical symbols and the keyboard to type the letters A, B, and C. (You can also enter the logical symbols from the keyboard by typing \&, |, and \neg, respectively. If you enter the logical symbols from the keyboard, make sure you add spaces before and after the binary connectives so that the columns under them will be reasonably spaced out.) If your sentence is well formed, the small “(1)” above the sentence will turn green.

2. To build the reference columns, click in the top left portion of the table to move your insertion point to the top of the first reference column. Enter C in this column. Then choose Add Column Before from the Table menu and enter B. Repeat this procedure and add a column headed by A. To fill in the reference columns, click under each of them in turn, and type the desired pattern of T’s and F’s.

3. Click under the various connectives in the target sentence, and notice that green squares appear in the columns whose values the connective depends upon. Select a column so that the highlighted columns are already filled.
in, and fill in that column with the appropriate truth values. Continue this process until your table is complete. When you are done use the Verify item from the Table menu to see if all the values are correct and your table complete. You can also verify your table using the colored button on the toolbar (just to the left of the print button). If you have filled the table correctly, green check marks should appear to the left of each row, and next to the target sentence. Red crosses indicate that you have made a mistake, and you should fix these now.

4. Once you have a correct and complete truth table, click on the Assessment button in the pink area under the toolbar. This will allow you to say whether you think the sentence is a tautology. Say that it is (since it is), and check your assessment by again selecting Verify from the Table menu (or by using the toolbar button). You should now see a green check mark next to the word “Tautology” on the assessment panel. Save your table as Table Tautology 1.

Congratulations

There is a slight problem with our definition of a tautology, in that it assumes that every sentence has a main connective. This is almost always the case, but not in sentences like:

\[ P \land Q \land R \]

For purposes of constructing truth tables, we will assume that the main connective in conjunctions with more than two conjuncts is always the rightmost \( \land \). That is to say, we will construct a truth table for \( P \land Q \land R \) the same way we would construct a truth table for:

\[ (P \land Q) \land R \]

More generally, we construct the truth table for:

\[ P_1 \land P_2 \land P_3 \land \ldots \land P_n \]

as if it were “punctuated” like this:

\[ (((P_1 \land P_2) \land P_3) \land \ldots) \land P_n \]

We treat long disjunctions similarly.

Any tautology is logically necessary. After all, its truth is guaranteed simply by its structure and the meanings of the truth-functional connectives. Tautologies are logical necessities in a very strong sense. Their truth is independent of both the way the world happens to be and even the meanings of the atomic sentences out of which they are composed.
It should be clear, however, that not all logically necessary claims are tautologies. The simplest example of a logically necessary claim that is not a tautology is the FOL sentence $a = a$. Since this is an atomic sentence, its truth table would contain one T and one F. The truth table method is too coarse to recognize that the row containing the F does not represent a genuine possibility.

You should be able to think of any number of sentences that are not tautological, but which nonetheless seem logically necessary. For example, the sentence

$$\neg(Larger(a, b) \land Larger(b, a))$$

cannot possibly be false, yet a truth table for the sentence will not show this. The sentence will be false in the row of the truth table that assigns T to both Larger(a, b) and Larger(b, a).

We now have two methods for exploring the notions of logical possibility and necessity, at least for the blocks language. First, there are the blocks worlds that can be constructed using Tarski’s World. If a sentence is true in
some such world, we have called it **tw**-possible. Similarly, if a sentence is true in every world in which it has a truth value (that is, in which its names all have referents), we can call it **tw**-necessary. The second method is that of truth tables. If a sentence comes out true in every row of its truth table, we could call it **tt**-necessary or, more traditionally, tautological. If a sentence is true in at least one row of its truth table, we will call it **tt**-possible.

None of these concepts correspond exactly to the vague notions of logical possibility and necessity. But there are clear and important relationships between the notions. On the necessity side, we know that all tautologies are logically necessary, and that all logical necessities are **tw**-necessary. These relationships are depicted in the “Euler circle” diagram in Figure 4.1, where we have represented the set of logical necessities as the interior of a circle with a fuzzy boundary. The set of tautologies is represented by a precise circle contained inside the fuzzy circle, and the set of Tarski’s World necessities is represented by a precise circle containing both these circles.

There is, in fact, another method for showing that a sentence is a logical truth, one that uses the technique of proofs. If you can prove a sentence using no premises whatsoever, then the sentence is logically necessary. In the following chapters, we will give you some more methods for giving proofs. Using these, you will be able to prove that sentences are logically necessary without constructing their truth tables. When we add quantifiers to our language, the gap between tautologies and logical truths will become very apparent, making the truth table method less useful. By contrast, the methods of proof that we discuss later will extend naturally to sentences containing quantifiers.

### Remember

Let $S$ be a sentence of **FOL** built up from atomic sentences by means of truth-functional connectives alone. A truth table for $S$ shows how the truth of $S$ depends on the truth of its atomic parts.

1. $S$ is a tautology if and only if every row of the truth table assigns **true** to $S$.

2. If $S$ is a tautology, then $S$ is a logical truth (that is, is logically necessary).

3. Some logical truths are not tautologies.

4. $S$ is **tt**-possible if and only if at least one row of the truth table assigns **true** to $S$. 

---

**Section 4.1**
Exercises

In this chapter, you will often be using Boole to construct truth tables. Although Boole has the capability of building and filling in reference columns for you, do not use this feature. To understand truth tables, you need to be able to do this yourself. In later chapters, we will let you use the feature, once you've learned how to do it yourself. The Grade Grinder will, by the way, be able to tell if Boole constructed the reference columns.

4.1 If you skipped the **You try it** section, go back and do it now. Submit the file Table Tautology 1.

4.2 Assume that A, B, and C are atomic sentences. Use Boole to construct truth tables for each of the following sentences and, based on your truth tables, say which are tautologies. Name your tables Table 4.2.x, where x is the number of the sentence.

1. \((A \land B) \lor (\neg A \lor \neg B)\)
2. \((A \land B) \lor (A \land \neg B)\)
3. \(\neg (A \land B) \lor C\)
4. \((A \lor B) \lor \neg (A \lor (B \land C))\)

4.3 In Exercise 4.2 you should have discovered that two of the four sentences are tautologies, and hence logical truths.

1. Suppose you are told that the atomic sentence A is in fact a logical truth (for example, \(a = a\)). Can you determine whether any additional sentences in the list (1)-(4) are logically necessary based on this information?
2. Suppose you are told that A is in fact a logically false sentence (for example, \(a \neq a\)). Can you determine whether any additional sentences in the list (1)-(4) are logical truths based on this information?

In the following four exercises, use Boole to construct truth tables and indicate whether the sentence is TT-possible and whether it is a tautology. Remember how you should treat long conjunctions and disjunctions.

4.4 \(\neg (B \land \neg C \land \neg B)\)

4.5 \(A \lor \neg (B \lor \neg (C \land A))\)

4.6 \(\neg [\neg A \lor (B \land C) \lor (A \land B)]\)

4.7 \(\neg [(\neg A \lor B) \land \neg (C \land D)]\)

4.8 Make a copy of the Euler circle diagram on page 102 and place the numbers of the following sentences in the appropriate region.

1. \(a = b\)
2. \(a = b \lor b = b\)
3. \( a = b \land b = b \)
4. \( \neg (\text{Large}(a) \land \text{Large}(b) \land \text{Adjoins}(a, b)) \)
5. \( \text{Larger}(a, b) \lor \neg \text{Larger}(a, b) \)
6. \( \text{Larger}(a, b) \lor \text{Smaller}(a, b) \)
7. \( \neg \text{Tet}(a) \lor \neg \text{Cube}(b) \lor a \neq b \)
8. \( \neg (\text{Small}(a) \land \text{Small}(b)) \lor \text{Small}(a) \)
9. \( \text{SameSize}(a, b) \lor \neg (\text{Small}(a) \land \text{Small}(b)) \)
10. \( \neg (\text{SameCol}(a, b) \land \text{SameRow}(a, b)) \)

4.9  
(Logical dependencies) Use Tarski’s World to open Weiner’s Sentences.

1. For each of the ten sentences in this file, construct a truth table in Boole and assess whether the sentence is \( \text{TT-possible} \). Name your tables \( \text{Table 4.9.x} \), where \( x \) is the number of the sentence in question. Use the results to fill in the first column of the following table:

<table>
<thead>
<tr>
<th>Sentence</th>
<th>TT-possible</th>
<th>TW-possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. In the second column of the table, put \( \text{yes} \) if you think the sentence is \( \text{TW-possible} \), that is, if it is possible to make the sentence true by building a world in Tarski’s World, and \( \text{no} \) otherwise. For each sentence that you mark \( \text{TW-possible} \), actually build a world in which it is true and name it \( \text{World 4.9.x} \), where \( x \) is the number of the sentence in question. The truth tables you constructed before may help you build these worlds.

3. Are any of the sentences \( \text{TT-possible} \) but not \( \text{TW-possible} \)? Explain why this can happen. Are any of the sentences \( \text{TW-possible} \) but not \( \text{TT-possible} \)? Explain why not.

Submit the files you created and turn in the table and explanations to your instructor.

4.10  
(Recommended for those who want to explore further)

Draw an Euler circle diagram similar to the diagram on page 102, but this time showing the relationship between the notions of logical possibility, \( \text{TW-possibility} \), and \( \text{TT-possibility} \). For each region in the diagram, indicate an example sentence that would fall in that region. Don’t forget the region that falls outside all the circles.

All necessary truths are obviously possible: since they are true in \( \text{all} \) possible circumstances, they are surely true in \( \text{some} \) possible circumstances. Given this reflection, where would the sentences from our previous diagram on page 102 fit into the new diagram?

4.11  
(Recommended for those who want to explore further)

Suppose that \( S \) is a tautology, with atomic sentences \( A, B, \) and \( C \). Suppose that we replace all occurrences of \( A \) by another sentence \( P \), possibly complex. Explain why the resulting sentence
is still a tautology. This is expressed by saying that substitution preserves tautologicality. Explain why substitution of atomic sentences does not always preserve logical truth, even though it preserves tautologies. Give an example.

Section 4.2

Logical and tautological equivalence

In the last chapter, we introduced the notion of logically equivalent sentences, sentences that have the same truth values in every possible circumstance. When two sentences are logically equivalent, we also say they have the same truth conditions, since the conditions under which they come out true or false are identical.

The notion of logical equivalence, like logical necessity, is somewhat vague, but not in a way that prevents us from studying it with precision. For here too we can introduce precise concepts that bear a clear relationship to the intuitive notion we aim to understand better. The key concept we will introduce in this section is that of tautological equivalence. Two sentences are tautologically equivalent if they can be seen to be equivalent simply in virtue of the meanings of the truth-functional connectives. As you might expect, we can check for tautological equivalence using truth tables.

Suppose we have two sentences, $S$ and $S'$, that we want to check for tautological equivalence. What we do is construct a truth table with a reference column for each of the atomic sentences that appear in either of the two sentences. To the right, we write both $S$ and $S'$, with a vertical line separating them, and fill in the truth values under the connectives as usual. We call this a joint truth table for the sentences $S$ and $S'$. When the joint truth table is completed, we compare the column under the main connective of $S$ with the column under the main connective of $S'$. If these columns are identical, then we know that the truth conditions of the two sentences are the same.

Let’s look at an example. Using $A$ and $B$ to stand for arbitrary atomic sentences, let us test the first DeMorgan law for tautological equivalence. We would do this by means of the following joint truth table.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>$\neg(A \land B)$</th>
<th>$\neg A \lor \neg B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

In this table, the columns in bold correspond to the main connectives of the
two sentences. Since these columns are identical, we know that the sentences must have the same truth values, no matter what the truth values of their atomic constituents may be. This holds simply in virtue of the structure of the two sentences and the meanings of the Boolean connectives. So, the two sentences are indeed tautologically equivalent.

Let’s look at a second example, this time to see whether the sentence \( \neg((A \lor B) \land \neg C) \) is tautologically equivalent to \((\neg A \land \neg B) \lor C\). To construct a truth table for this pair of sentences, we will need eight rows, since there are three atomic sentences. The completed table looks like this.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>( \neg((A \lor B) \land \neg C) )</th>
<th>((\neg A \land \neg B) \lor C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Once again, scanning the final columns under the two main connectives reveals that the sentences are tautologically equivalent, and hence logically equivalent.

All tautologically equivalent sentences are logically equivalent, but the reverse does not in general hold. Indeed, the relationship between these notions is the same as that between tautologies and logical truths. Tautological equivalence is a strict form of logical equivalence, one that won’t apply to some logically equivalent pairs of sentences. Consider the pair of sentences:

\[
\begin{align*}
  a &= b \land \text{Cube}(a) \\
  a &= b \land \text{Cube}(b)
\end{align*}
\]

These sentences are logically equivalent, as is demonstrated in the following informal proof.

**Proof:** Suppose that the sentence \( a = b \land \text{Cube}(a) \) is true. Then \( a = b \) and \( \text{Cube}(a) \) are both true. Using the indiscernibility of identi- cals (Identity Elimination), we know that \( \text{Cube}(b) \) is true, and hence that \( a = b \land \text{Cube}(b) \) is true. So the truth of \( a = b \land \text{Cube}(a) \) logically implies the truth of \( a = b \land \text{Cube}(b) \).

The reverse holds as well. For suppose that \( a = b \land \text{Cube}(b) \) is true. Then by symmetry of identity, we also know \( b = a \). From this and \( \text{Cube}(b) \) we can conclude \( \text{Cube}(a) \), and hence that \( a = b \land \text{Cube}(a) \)
is true. So the truth of $a = b \land \text{Cube}(b)$ implies the truth of $a = b \land \text{Cube}(a)$.

Thus $a = b \land \text{Cube}(a)$ is true if and only if $a = b \land \text{Cube}(b)$ is true.

This proof shows that these two sentences have the same truth values in any possible circumstance. For if one were true and the other false, this would contradict the conclusion of one of the two parts of the proof. But consider what happens when we construct a joint truth table for these sentences. Three atomic sentences appear in the pair of sentences, so the joint table will look like this. (Notice that the ordinary truth table for either of the sentences alone would have only four rows, but that the joint table must have eight. Do you understand why?)

<table>
<thead>
<tr>
<th>$a = b$</th>
<th>$\text{Cube}(a)$</th>
<th>$\text{Cube}(b)$</th>
<th>$a = b \land \text{Cube}(a)$</th>
<th>$a = b \land \text{Cube}(b)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

This table shows that the two sentences are not tautologically equivalent, since it assigns the sentences different values in the second and third rows. Look closely at those two rows to see what’s going on. Notice that in both of these rows, $a = b$ is assigned T while $\text{Cube}(a)$ and $\text{Cube}(b)$ are assigned different truth values. Of course, we know that neither of these rows corresponds to a logically possible circumstance, since if $a$ and $b$ are identical, the truth values of $\text{Cube}(a)$ and $\text{Cube}(b)$ must be the same. But the truth table method doesn’t detect this, since it is sensitive only to the meanings of the truth-functional connectives.

As we expand our language to include quantifiers, we will find many logical equivalences that are not tautological equivalences. But this is not to say there aren’t a lot of important and interesting tautological equivalences. We’ve already highlighted three in the last chapter: double negation and the two DeMorgan equivalences. We leave it to you to check that these principles are, in fact, tautological equivalences. In the next section, we will introduce other principles and see how they can be used to simplify sentences of FOL.
Remember

Let $S$ and $S'$ be a sentences of FOL built up from atomic sentences by means of truth-functional connectives alone. To test for tautological equivalence, we construct a joint truth table for the two sentences.

1. $S$ and $S'$ are tautologically equivalent if and only if every row of the joint truth table assigns the same values to $S$ and $S'$.
2. If $S$ and $S'$ are tautologically equivalent, then they are logically equivalent.
3. Some logically equivalent sentences are not tautologically equivalent.

Exercises

In Exercises 4.12-4.18, use Boole to construct joint truth tables showing that the pairs of sentences are logically (indeed, tautologically) equivalent. To add a second sentence to your joint truth table, choose Add Column After from the Table menu. Don’t forget to specify your assessments, and remember, you should build and fill in your own reference columns.

4.12 (DeMorgan)

\( \neg (A \lor B) \) and \( \neg A \neg B \)

4.13 (Associativity)

\( (A \land B) \land C \) and \( A \land (B \land C) \)

4.14 (Associativity)

\( (A \lor B) \lor C \) and \( A \lor (B \lor C) \)

4.15 (Idempotence)

\( A \land B \land A \) and \( A \land B \)

4.16 (Idempotence)

\( A \lor B \lor A \) and \( A \lor B \)

4.17 (Distribution)

\( A \land (B \lor C) \) and \( (A \land B) \lor (A \land C) \)

4.18 (Distribution)

\( A \lor (B \land C) \) and \( (A \lor B) \land (A \lor C) \)

4.19 (tw-equivalence) Suppose we introduced the notion of tw-equivalence, saying that two sentences of the blocks language are tw-equivalent if and only if they have the same truth value in every world that can be constructed in Tarski’s World.

1. What is the relationship between tw-equivalence, tautological equivalence and logical equivalence?
2. Give an example of a pair of sentences that are tw-equivalent but not logically equivalent.
Our main concern in this book is with the logical consequence relation, of which logical truth and logical equivalence can be thought of as very special cases: A logical truth is a sentence that is a logical consequence of any set of premises, and logically equivalent sentences are sentences that are logical consequences of one another.

As you’ve probably guessed, truth tables allow us to define a precise notion of tautological consequence, a strict form of logical consequence, just as they allowed us to define tautologies and tautological equivalence, strict forms of logical truth and logical equivalence.

Let’s look at the simple case of two sentences, P and Q, both built from atomic sentences by means of truth-functional connectives. Suppose you want to know whether Q is a consequence of P. Create a joint truth table for P and Q, just like you would if you were testing for tautological equivalence. After you fill in the columns for P and Q, scan the columns under the main connectives for these sentences. In particular, look at every row of the table in which P is true. If each such row is also one in which Q is true, then Q is said to be a tautological consequence of P. The truth table shows that if P is true, then Q must be true as well, and that this holds simply due to the meanings of the truth-functional connectives.

Just as tautologies are logically necessary, so too any tautological consequence Q of a sentence P must also be a logical consequence of P. We can see this by proving that if Q is not a logical consequence of P, then it can’t possibly pass our truth table test for tautological consequence.

**Proof:** Suppose Q is not a logical consequence of P. Then by our definition of logical consequence, there must be a possible circumstance in which P is true but Q is false. This circumstance will determine truth values for the atomic sentences in P and Q, and these values will correspond to a row in the joint truth table for P and Q, since all possible assignments of truth values to the atomic sentences are represented in the truth table. Further, since P and Q are built up from the atomic sentences by truth-functional connectives, and since the former is true in the original circumstance and the latter false, P will be assigned T in this row and Q will be assigned F. Hence, Q is not a tautological consequence of P.

Let’s look at a very simple example. Suppose we wanted to check to see whether A ∨ B is a consequence of A ∧ B. The joint truth table for these sen-
Logical and tautological consequence

The truth table method of checking tautological consequence is not restricted to just one premise. You can apply it to arguments with any number of premises \( P_1, \ldots, P_n \) and conclusion \( Q \). To do so, you have to construct a joint truth table for all of the sentences \( P_1, \ldots, P_n \) and \( Q \). Once you've done this, you need to check every row in which the premises all come out true to see whether the conclusion comes out true as well. If so, the conclusion is a tautological consequence of the premises.

Let's try this out on a couple of simple examples. First, suppose we want to check to see whether \( B \) is a consequence of the two premises \( A \lor B \) and \( \neg A \). The joint truth table for these three sentences comes out like this. (Notice that since one of our target sentences, the conclusion \( B \), is atomic, we have simply repeated the reference column when this sentence appears again on
the right.)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A ∨ B</th>
<th>¬A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

Scanning the columns under our two premises, A ∨ B and ¬A, we see that there is only one row where both premises come out true, namely the third. And in the third row, the conclusion B also comes out true. So B is indeed a tautological (and hence logical) consequence of these premises.

In both of the examples we’ve looked at so far, there has been only one row in which the premises all came out true. This makes the arguments easy to check for validity, but it’s not at all something you can count on. For example, suppose we used the truth table method to check whether A ∨ C is a consequence of A ∨ ¬B and B ∨ C. The joint truth table for these three sentences looks like this.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>A ∨ ¬B</th>
<th>B ∨ C</th>
<th>A ∨ C</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T  F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T  F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T  T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T  T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F  F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F  F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T  T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T  T</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Here, there are four rows in which the premises, A ∨ ¬B and B ∨ C, are both true: the first, second, third, and seventh. But in each of these rows the conclusion, A ∨ C, is also true. The conclusion is true in other rows as well, but we don’t care about that. This inference, from A ∨ ¬B and B ∨ C to A ∨ C, is logically valid, and is an instance of an important pattern known in computer science as resolution.

We should look at an example where the truth table method reveals that the conclusion is not a tautological consequence of the premises. Actually, the last truth table will serve this purpose. For this table also shows that the sentence A ∨ ¬B is not a tautological consequence of the two premises B ∨ C and A ∨ C. Can you find the row that shows this? (Hint: It’s got to be the first, second, third, fifth, or seventh, since these are the rows in which B ∨ C and A ∨ C are both true.)
Remember

Let $P_1, \ldots, P_n$ and $Q$ be sentences of $\text{FOL}$ built up from atomic sentences by means of truth functional connectives alone. Construct a joint truth table for all of these sentences.

1. $Q$ is a tautological consequence of $P_1, \ldots, P_n$ if and only if every row that assigns $T$ to each of $P_1, \ldots, P_n$ also assigns $T$ to $Q$.

2. If $Q$ is a tautological consequence of $P_1, \ldots, P_n$, then $Q$ is also a logical consequence of $P_1, \ldots, P_n$.

3. Some logical consequences are not tautological consequences.

Exercises

For each of the arguments below, use the truth table method to determine whether the conclusion is a tautological consequence of the premises. Your truth table for Exercise 4.24 will be fairly large. It’s good for the soul to build a large truth table every once in a while. Be thankful you have Boole to help you. (But make sure you build your own reference columns!)

4.20

\[
\begin{array}{c}
\text{(Tet(a) } \land \text{ Small(a)) } \lor \text{ Small(b)} \\
\text{Small(a) } \lor \text{ Small(b)}
\end{array}
\]

4.21

\[
\begin{array}{c}
\text{Taller(claire, max) } \lor \text{ Taller(max, claire)} \\
\text{Taller(claire, max)} \\
\neg\text{Taller(max, claire)}
\end{array}
\]

4.22

\[
\begin{array}{c}
\text{Large(a)} \\
\text{Cube(a) } \lor \text{ Dodec(a)} \\
\text{(Cube(a) } \land \text{ Large(a)) } \lor \text{ (Dodec(a) } \land \text{ Large(a))}
\end{array}
\]

4.23

\[
\begin{array}{c}
A \lor \neg B \\
B \lor C \\
C \lor D \\
A \lor \neg D
\end{array}
\]

4.24

\[
\begin{array}{c}
\neg A \lor B \lor C \\
\neg C \lor D \\
\neg(B \land \neg E) \\
D \lor \neg A \lor E
\end{array}
\]

4.25

Give an example of two different sentences $A$ and $B$ in the blocks language such that $A \land B$ is a logical consequence of $A \lor B$. [Hint: Note that $A \land A$ is a logical consequence of $A \lor A$, but here we insist that $A$ and $B$ be distinct sentences.]
Section 4.4

Tautological consequence in Fitch

We hope you solved Exercise 4.24, because the solution gives you a sense of both the power and the drawbacks of the truth table method. We were tempted to ask you to construct a table requiring 64 rows, but thought better of it. Constructing large truth tables may build character, but like most things that build character, it’s a drag.

Checking to see if \( Q \) is a tautological consequence of \( P_1, \ldots, P_n \) is a mechanical procedure. If the sentences are long it may require a lot of tedious work, but it doesn’t take any originality. This is just the sort of thing that computers are good at. Because of this, we have built a mechanism into Fitch, called Taut Con, that is similar to Ana Con but checks to see whether a sentence is a tautological consequence of the sentences cited in support. Like Ana Con, Taut Con is not really an inference rule (we will introduce inference rules for the Boolean connectives in Chapter 6), but is useful for quickly testing whether one sentence follows tautologically from others.

You try it

1. Launch Fitch and open the file Taut Con 1. In this file you will find an argument that has the same form as the argument in Exercise 4.23. (Ignore the two goal sentences. We’ll get to them later.) Move the focus slider to the last step of the proof. From the Rule? menu, go down to the Con submenu and choose Taut Con.

2. Now cite the three premises as support for this sentence and check the step. The step will not check out since this sentence is not a tautological consequence of the premises, as you discovered if you did Exercise 4.23, which has the same form as this inference.

3. Edit the step that did not check out to read:

\[
\text{Home}(\text{max}) \lor \text{Home}(\text{carl})
\]

This sentence is a tautological consequence of two of the premises. Figure out which two and cite just them. If you cited the right two, the step should check out. Try it.

4. Add another step to the proof and enter the sentence:

\[
\text{Home}(\text{carl}) \lor (\text{Home}(\text{max}) \land \text{Home}(\text{pris}))
\]
Use **Taut Con** to see if this sentence follows tautologically from the three premises. Choose **Verify Proof** from the **Proof** menu. You will find that although the step checks out, the goal does not. This is because we have put a special constraint on your use of **Taut Con** in this exercise.

5. Choose **View Goal Constraints** from the **Goal** menu. You will find that in this proof, you are allowed to use **Taut Con**, but can only cite two or fewer support sentences when you use it. Close the goal window to get back to the proof.

6. The sentence you entered also follows from the sentence immediately above it plus just one of the three premises. Uncite the three premises and see if you can get the step to check out citing just *two* sentences in support. Once you succeed, verify the proof and save it as **Proof Taut Con 1**. Do not close the proof, since it will be needed in the next “You Try It”.

**Congratulations**

You are probably curious about the relationship between **Taut Con** and **Ana Con**—and for that matter, what the other mysterious item on the **Con** menu, **FO Con**, might do. These are in fact three increasingly strong methods that Fitch uses to test for logical consequence. **Taut Con** is the weakest. It checks to see whether the current step follows from the cited sentences in virtue of the meanings of the truth-functional connectives. It ignores the meanings of any predicates that appear in the sentence and, when we introduce quantifiers into the language, it will ignore those as well.

To help you keep track of the information that **Taut Con** considers when checking a step, Fitch has special “goggles” which obscure the information that will not be considered when checking the step. As we said above, the only things that matter when checking a step justified by **Taut Con** step are the propositional connectives in the formulae, and the pattern of occurrence of the atomic formulae.

What this means is that the meanings of the predicate symbols and names in the formulae do not matter, and Fitch’s goggles obscure this information. When you put the goggles on, every individual atomic formula involved in the step appears as a block of color, hiding the particular atomic formula that is present. Every occurrence of the same atomic formula will be represented by the same color, and different formulae will have different colors.

**You try it**

1. Return to the file **Taut Con 1** again that you made in the previous “You Try It” section, and focus on the last step of the proof, which contains an
application of the Taut Con rule.

2. Click on the picture of a pair of goggles that appears to the right of the rule name, and notice how the conclusion and the cited sentences change into blocks of color.

3. You should see a more colorful version of this

```
[□ purple □] ∨ [□ yellow □]
[□ green □] ∨ [□ purple □]
[□ purple □] ∨ ([□ green □] ∧ [□ yellow □])
```

Be sure to understand how the atomic formulas relate to their corresponding colors.

4. There is nothing to save.

.......................................................... Congratulations

**FO Con**, which stands for “first-order consequence,” pays attention to the truth-functional connectives, the quantifiers, and the identity predicate when it checks for consequence. **FO Con** would, for example, identify $a = c$ as a consequence of $a = b \land b = c$. It is stronger than **Taut Con** in the sense that any consequence that **Taut Con** recognizes as valid will also be recognized by **FO Con**. But it may take longer since it has to apply a more complex procedure, thanks to identity and the quantifiers. After we get to quantifiers, we’ll talk more about the procedure it is applying.

The strongest rule of the three is **Ana Con**, which tries to recognize consequences due to truth-functional connectives, quantifiers, identity, and most of the blocks language predicates. (**Ana Con** ignores Between and Adjoins, simply for practical reasons.) Any inference that checks out using either **Taut Con** or **FO Con** should, in principle, check out using **Ana Con** as well. In practice, though, the procedure that **Ana Con** uses may bog down or run out of memory in cases where the first two have no trouble.

As we said before, you should only use a procedure from the **Con** menu when the exercise makes clear that the procedure is allowed in the solution. Moreover if an exercise asks you to use **Taut Con**, don’t use **FO Con** or **Ana Con** instead, even if these more powerful rules seem to work just as well. If you are in doubt about which rules you are allowed to use, choose **View Goal Constraints** from the **Goal** menu.
You try it

1. Open the file Taut Con 2. You will find a proof containing ten steps whose rules have not been specified.

2. Focus on each step in turn. You will find that the supporting steps have already been cited. Convince yourself that the step follows from the cited sentences. Is it a tautological consequence of the sentences cited? If so, change the rule to Taut Con and see if you were right. If not, change it to Ana Con and see if it checks out. (If Taut Con will work, make sure you use it rather than the stronger Ana Con.)

3. When all of your steps check out using Taut Con or Ana Con, go back and find the one step whose rule can be changed from Ana Con to the weaker FO Con.

4. When each step checks out using the weakest Con rule possible, save your proof as Proof Taut Con 2.

Congratulations

Exercises

4.26 If you skipped the You try it sections, go back and do them now. Submit the files Proof Taut Con 1 and Proof Taut Con 2.

For each of the following arguments, decide whether the conclusion is a tautological consequence of the premises. If it is, submit a proof that establishes the conclusion using one or more applications of Taut Con. Do not cite more than two sentences at a time for any of your applications of Taut Con. If the conclusion is not a consequence of the premises, submit a counterexample world showing that the argument is not valid.

4.27

| Cube(a) ∨ Cube(b) |
| Dodec(c) ∨ Dodec(d) |
| ¬Cube(a) ∨ ¬Dodec(c) |
| Cube(b) ∨ Dodec(d) |

4.28

| Large(a) ∨ Large(b) |
| Large(a) ∨ Large(c) |
| Large(a) ∧ (Large(b) ∨ Large(c)) |
### Section 4.5

**Pushing negation around**

When two sentences are logically equivalent, each is a logical consequence of the other. As a result, in giving an informal proof, you can always go from an established sentence to one that is logically equivalent to it. This fact makes observations like the DeMorgan laws and double negation quite useful in giving informal proofs.

What makes these equivalences even more useful is the fact that logically equivalent sentences can be substituted for one another in the context of a larger sentence and the resulting sentences will also be logically equivalent. An example will help illustrate what we mean. Suppose we start with the sentence:

\[ \neg(Cube(a) \land \neg Small(a)) \]

By the principle of double negation, we know that \( Small(a) \) is logically equivalent to \( \neg\neg Small(a) \). Since these have exactly the same truth conditions, we can substitute \( Small(a) \) for \( \neg\neg Small(a) \) in the context of the above sentence, and the result,

\[ \neg(Cube(a) \land Small(a)) \]

will be logically equivalent to the original, a fact that you can check by constructing a joint truth table for the two sentences.

We can state this important fact in the following way. Let’s write \( S(P) \) for an FOL sentence that contains the (possibly complex) sentence \( P \) as a component part, and \( S(Q) \) for the result of substituting \( Q \) for \( P \) in \( S(P) \). Then if \( P \) and \( Q \) are logically equivalent:

\[ P \leftrightarrow Q \]

it follows that \( S(P) \) and \( S(Q) \) are also logically equivalent:
S(P) ⇔ S(Q)

This is known as the principle of substitution of logical equivalents.

We won’t prove this principle at the moment, because it requires a proof by induction, a style of proof we get to in a later chapter. But the observation allows us to use a few simple equivalences to do some pretty amazing things. For example, using only the two DeMorgan laws and double negation, we can take any sentence built up with ∧, ∨, and ¬, and transform it into one where ¬ applies only to atomic sentences. Another way of expressing this is that any sentence built out of atomic sentences using the three connectives ∧, ∨, and ¬ is logically equivalent to one built from literals using just ∧ and ∨.

To obtain such a sentence, you simply drive the ¬ in, switching ∧ to ∨, ∨ to ∧, and canceling any pair of ¬’s that are right next to each other, not separated by any parentheses. Such a sentence is said to be in negation normal form or NNF. Here is an example of a derivation of the negation normal form of a sentence. We use A, B, and C to stand for any atomic sentences of the language.

\[
\neg((A \lor B) \land \neg C) \equiv \neg(A \lor B) \lor \neg \neg C \\
\equiv \neg(A \lor B) \lor C \\
\equiv (\neg A \land \neg B) \lor C
\]

In reading and giving derivations of this sort, remember that the symbol ⇔ is not itself a symbol of the first-order language, but a shorthand way of saying that two sentences are logically equivalent. In this derivation, the first step is an application of the first DeMorgan law to the whole sentence. The second step applies double negation to the component ¬¬C. The final step is an application of the second DeMorgan law to the component ¬(A ∨ B). The sentence we end up with is in negation normal form, since the negation signs apply only to atomic sentences.

We end this section with a list of some additional logical equivalences that allow us to simplify sentences in useful ways. You already constructed truth tables for most of these equivalences in Exercises 4.13-4.16 at the end of Section 4.2.

1. (Associativity of ∧) An FOL sentence P ∧ (Q ∧ R) is logically equivalent to (P ∧ Q) ∧ R, which is in turn equivalent to P ∧ Q ∧ R. That is, associativity

\[
P ∧ (Q ∧ R) \equiv (P ∧ Q) ∧ R \equiv P ∧ Q ∧ R
\]

2. (Associativity of ∨) An FOL sentence P ∨ (Q ∨ R) is logically equivalent to (P ∨ Q) ∨ R, which is in turn equivalent to P ∨ Q ∨ R. That is, associativity

\[
P ∨ (Q ∨ R) \equiv (P ∨ Q) ∨ R \equiv P ∨ Q ∨ R
\]
3. (Commutativity of $\land$) A conjunction $P \land Q$ is logically equivalent to $Q \land P$. That is,

$$P \land Q \leftrightarrow Q \land P$$

As a result, any rearrangement of the conjuncts of an FOL sentence is logically equivalent to the original. For example, $P \land Q \land R$ is equivalent to $R \land Q \land P$.

4. (Commutativity of $\lor$) A disjunction $P \lor Q$ is logically equivalent to $Q \lor P$. That is,

$$P \lor Q \leftrightarrow Q \lor P$$

As a result, any rearrangement of the disjuncts of an FOL sentence is logically equivalent to the original. For example, $P \lor Q \lor R$ is equivalent to $R \lor Q \lor P$.

5. (Idempotence of $\land$) A conjunction $P \land P$ is equivalent to $P$. That is,

$$P \land P \leftrightarrow P$$

More generally (given Commutativity), any conjunction with a repeated conjunct is equivalent to the result of removing all but one occurrence of that conjunct. For example, $P \land Q \land P$ is equivalent to $P \land Q$.

6. (Idempotence of $\lor$) A disjunction $P \lor P$ is equivalent to $P$. That is,

$$P \lor P \leftrightarrow P$$

More generally (given Commutativity), any disjunction with a repeated disjunct is equivalent to the result of removing all but one occurrence of that disjunct. For example, $P \lor Q \lor P$ is equivalent to $P \lor Q$.

Here is an example where we use some of these laws to show that the first sentence in the following list is logically equivalent to the last. Once again (as in what follows), we use $A$, $B$, and $C$ to stand for arbitrary atomic sentences of FOL. Thus the result is in negation normal form.

$$(A \lor B) \land C \land (\neg B \land \neg A) \lor B) \leftrightarrow (A \lor B) \land C \land ((\neg B \lor \neg B) \lor A) \land B)$$

$$\leftrightarrow (A \lor B) \land C \land ((B \lor A) \lor B)$$

$$\leftrightarrow (A \lor B) \land C \land (B \lor B)$$

$$\leftrightarrow (A \lor B) \land C \land (B \lor A)$$

$$\leftrightarrow (A \lor B) \land C \land (A \lor B)$$

$$\leftrightarrow (A \lor B) \land C$$
We call a demonstration of this sort a *chain of equivalences*. The first step in this chain is justified by one of the DeMorgan laws. The second step involves two applications of double negation. In the next step we use associativity to remove the unnecessary parentheses. In the fourth step, we use idempotence of $\lor$. The next to the last step uses commutativity of $\lor$, while the final step uses idempotence of $\land$.

**Remember**

1. *Substitution of equivalents*: If $P$ and $Q$ are logically equivalent:

   $$ P \leftrightarrow Q $$

   then the results of substituting one for the other in the context of a larger sentence are also logically equivalent:

   $$ S(P) \leftrightarrow S(Q) $$

2. A sentence is in *negation normal form* (NNF) if all occurrences of $\neg$ apply directly to atomic sentences.

3. Any sentence built from atomic sentences using just $\land$, $\lor$, and $\neg$ can be put into negation normal form by repeated application of the De-Morgan laws and double negation.

4. Sentences can often be further simplified using the principles of associativity, commutativity, and idempotence.

**Exercises**

**4.31** (Negation normal form) Use Tarski’s World to open Turing’s Sentences. You will find the following five sentences, each followed by an empty sentence position.

1. $\neg(Cube(a) \land Larger(a, b))$
2. $\neg(Cube(a) \lor \neg Larger(b, a))$
3. $\neg(\neg Cube(a) \lor \neg Larger(a, b) \lor a \neq b)$
4. $\neg(Tet(b) \lor (Large(c) \land \neg Smaller(d, e)))$
5. $\neg(Tet(b) \lor \neg Tet(f) \lor \neg Dodec(f))$

In the empty positions, write the negation normal form of the sentence above it. Then build any world where all of the names are in use. If you have gotten the negation normal forms
correct, each even numbered sentence will have the same truth value in your world as the odd numbered sentence above it. Verify that this is so in your world. Submit the modified sentence file as Sentences 4.31.

4.32 (Negation normal form) Use Tarski’s World to open the file Sextus’ Sentences. In the odd numbered slots, you will find the following sentences.

1. \( \neg(\text{Home(carl)} \land \neg\text{Home(claire)}) \)
2. \( \neg[\text{Happy(max)} \land (\neg\text{Likes(carl, claire)} \lor \neg\text{Likes(claire, carl)})] \)
3. \( \neg\neg\neg[(\text{Home(max)} \lor \text{Home(carl)}) \land (\text{Happy(max)} \lor \text{Happy(carl)})] \)

Use Double Negation and DeMorgan’s laws to put each sentence into negation normal form in the slot below it. Submit the modified file as Sentences 4.32.

In each of the following exercises, use associativity, commutativity, and idempotence to simplify the sentence as much as you can using just these rules. Your answer should consist of a chain of logical equivalences like the chain given on page 120. At each step of the chain, indicate which principle you are using.

4.33 \( (A \land B) \land A \)
4.34 \( (B \land (A \land B \land C)) \)

4.35 \( (A \lor B) \lor (C \land D) \lor A \)
4.36 \( (\neg A \lor B) \lor (B \lor C) \)

4.37 \( (A \land B) \lor C \lor (B \lor A) \lor A \)

Section 4.6

Conjunctive and disjunctive normal forms

We have seen that with a few simple principles of Boolean logic, we can start with a sentence and transform it into a logically equivalent sentence in negation normal form, one where all negations occur in front of atomic sentences. We can improve on this by introducing the so-called distributive laws. These additional equivalences will allow us to transform sentences into what are known as conjunctive normal form (CNF) and disjunctive normal form (DNF). These normal forms are quite important in certain applications of logic in computer science, as we discuss in Chapter 17. We will also use disjunctive normal form to demonstrate an important fact about the Boolean connectives in Chapter 7.

Recall that in algebra you learned that multiplication distributes over addition: \( a \times (b + c) = (a \times b) + (a \times c) \). The distributive laws of logic look formally
much the same. One version tells us that $P \land (Q \lor R)$ is logically equivalent to $(P \land Q) \lor (P \land R)$. That is, $\land$ distributes over $\lor$. To see that this is so, notice that the first sentence is true if and only if $P$ plus at least one of $Q$ or $R$ is true. But a moment’s thought shows that the second sentence is true in exactly the same circumstances. This can also be confirmed by constructing a joint truth table for the two sentences, which you’ve already done if you did Exercise 4.17.

In arithmetic, $+$ does not distribute over $\times$. However, $\lor$ does distribute over $\land$. That is to say, $P \lor (Q \land R)$ is logically equivalent to $(P \lor Q) \land (P \lor R)$, as you also discovered in Exercise 4.18.

<table>
<thead>
<tr>
<th>Remember</th>
</tr>
</thead>
<tbody>
<tr>
<td>(The distributive laws) For any sentences $P$, $Q$, and $R$:</td>
</tr>
<tr>
<td>1. Distribution of $\land$ over $\lor$: $P \land (Q \lor R) \iff (P \land Q) \lor (P \land R)$</td>
</tr>
<tr>
<td>2. Distribution of $\lor$ over $\land$: $P \lor (Q \land R) \iff (P \lor Q) \land (P \lor R)$</td>
</tr>
</tbody>
</table>

As you may recall from algebra, the distributive law for $\times$ over $+$ is incredibly useful. It allows us to transform any algebraic expression involving $+$ and $\times$, no matter how complex, into one that is just a sum of products. For example, the following transformation uses distribution three times.

\[
(a + b)(c + d) = (a + b)c + (a + b)d \\
= ac + bc + (a + b)d \\
= ac + bc + ad + bd
\]

In exactly the same way, the distribution of $\land$ over $\lor$ allows us to transform any sentence built up from literals by means of $\land$ and $\lor$ into a logically equivalent sentence that is a disjunction of (one or more) conjunctions of (one or more) literals. That is, using this first distributive law, we can turn any sentence in negation normal form into a sentence that is a disjunction of conjunctions of literals. A sentence in this form is said to be in disjunctive normal form (DNF).

Here is an example that parallels our algebraic example. Notice that, as in the algebraic example, we are distributing in from the right as well as the left, even though our statement of the rule only illustrates distribution from the left.

\[
(A \lor B) \land (C \lor D) \iff [(A \lor B) \land C] \lor [(A \lor B) \land D] \\
\iff (A \land C) \lor (B \land C) \lor [(A \lor B) \land D] \\
\iff (A \land C) \lor (B \land C) \lor (A \land D) \lor (B \land D)
\]
As you can see, distribution of $\land$ over $\lor$ lets us drive conjunction signs deeper and deeper, just as the DeMorgan laws allow us to move negations deeper. Thus, if we take any sentence and first use DeMorgan (and double negation) to get a sentence in negation normal form, we can then use this first distribution law to get a sentence in disjunctive normal form, one in which all the conjunction signs apply to literals.

Likewise, using distribution of $\lor$ over $\land$, we can turn any negation normal form sentence into one that is a conjunction of one or more sentences, each of which is a disjunction of one or more literals. A sentence in this form is said to be in conjunctive normal form (CNF). Here’s an example, parallel to the one given above but with $\land$ and $\lor$ interchanged:

$$(A \land B) \lor (C \land D) \iff [(A \land B) \lor C] \land [(A \land B) \lor D]$$
$$\iff (A \lor C) \land (B \lor C) \land [(A \land B) \lor D]$$
$$\iff (A \lor C) \land (B \lor C) \land (A \lor D) \land (B \lor D)$$

On page 119, we showed how to transform the sentence $\neg((A \lor B) \land \neg C)$ into one in negation normal form. The result was $(\neg A \land \neg B) \lor C$. This sentence just happens to be in disjunctive normal form. Let us repeat our earlier transformation, but continue until we get a sentence in conjunctive normal form.

$$\neg((A \lor B) \land \neg C) \iff \neg(A \lor B) \lor \neg\neg C$$
$$\iff \neg(A \lor B) \lor C$$
$$\iff (\neg A \land \neg B) \lor C$$
$$\iff (\neg A \lor C) \land (\neg B \lor C)$$

It is important to remember that a sentence can count as being in both conjunctive and disjunctive normal forms at the same time. For example, the sentence

$$\text{Home(claire)} \land \neg\text{Home(max)}$$

is in both DNF and CNF. On the one hand, it is in disjunctive normal form since it is a disjunction of one sentence (itself) which is a conjunction of two literals. On the other hand, it is in conjunctive normal form since it is a conjunction of two sentences, each of which is a disjunction of one literal.

In case you find this last remark confusing, here are simple tests for whether sentences are in disjunctive normal form and conjunctive normal form. The tests assume that the sentence has no unnecessary parentheses and contains only the connectives $\land$, $\lor$, and $\neg$.

To check whether a sentence is in DNF, ask yourself whether all the
negation signs apply directly to atomic sentences and whether all the conjunction signs apply directly to literals. If both answers are yes, then the sentence is in disjunctive normal form.

To check whether a sentence is in CNF, ask yourself whether all the negation signs apply directly to atomic sentences and all the disjunction signs apply directly to literals. If both answers are yes, then the sentence is in conjunctive normal form.

Now look at the above sentence again and notice that it passes both of these tests (in the CNF case because it has no disjunction signs).

Remember

1. A sentence is in disjunctive normal form (DNF) if it is a disjunction of one or more conjunctions of one or more literals.

2. A sentence is in conjunctive normal form (CNF) if it is a conjunction of one or more disjunctions of one or more literals.

3. Distribution of $\land$ over $\lor$ allows you to transform any sentence in negation normal form into disjunctive normal form.

4. Distribution of $\lor$ over $\land$ allows you to transform any sentence in negation normal form into conjunctive normal form.

5. Some sentences are in both CNF and DNF.

You try it

1. Use Tarski’s World to open the file DNF Example. In this file you will find two sentences. The second sentence is the result of putting the first into disjunctive normal form, so the two sentences are logically equivalent.

2. Build a world in which the sentences are true. Since they are equivalent, you could try to make either one true, but you will find the second one easier to work on.

3. Play the game for each sentence, committed correctly to the truth of the sentence. You should be able to win both times. Count the number of steps it takes you to win.
4. In general it is easier to evaluate the truth value of a sentence in disjunctive normal form. This comes out in the game, which takes at most three steps for a sentence in DNF, one each for \( \lor \), \( \land \), and \( \neg \), in that order. There is no limit to the number of steps a sentence in other forms may take.

5. Save the world you have created as World DNF 1. Congratulations

Exercises

4.38 If you skipped the You try it section, go back and do it now. Submit the file World DNF 1.

4.39 Open CNF Sentences. In this file you will find the following conjunctive normal form sentences in the odd numbered positions, but you will see that the even numbered positions are blank.

1. \((\text{LeftOf}(a,b) \lor \text{BackOf}(a,b)) \land \text{Cube}(a)\)
2. \((\text{Larger}(a,b) \land (\text{Cube}(a) \lor \text{Tet}(a) \lor a = b))\)
3. \((\text{Between}(a,b,c) \lor \text{Tet}(a) \lor \neg \text{Tet}(b)) \land \text{Dodec}(c)\)
4. \((\text{Cube}(a) \land \text{Cube}(b) \land (\neg \text{Small}(a) \lor \neg \text{Small}(b))\)
5. \((\text{Small}(a) \lor \text{Medium}(a)) \land (\text{Cube}(a) \lor \neg \text{Dodec}(a))\)

In the even numbered positions you should fill in a DNF sentence logically equivalent to the sentence above it. Check your work by opening several worlds and checking to see that each of your sentences has the same truth value as the one above it. Submit the modified file as Sentences 4.39.

4.40 Open More CNF Sentences. In this file you will find the following sentences in every third position.

1. \(\neg[(\text{Cube}(a) \land \neg \text{Small}(a)) \lor (\neg \text{Cube}(a) \land \text{Small}(a))]\)
2. \(\neg[(\text{Cube}(a) \lor \neg \text{Small}(a)) \land (\neg \text{Cube}(a) \lor \text{Small}(a))]\)
3. \(\neg(\text{Cube}(a) \land \text{Larger}(a,b)) \land \text{Dodec}(b)\)
4. \(\neg(\neg \text{Cube}(a) \land \text{Tet}(b))\)
5. \(\neg \text{Cube}(a) \lor \text{Tet}(b)\)

The two blanks that follow each sentence are for you to first transform the sentence into negation normal form, and then put that sentence into CNF. Again, check your work by opening several worlds to see that each of your sentences has the same truth value as the original. When you are finished, submit the modified file as Sentences 4.40.
In Exercises 4.41-4.43, use a chain of equivalences to convert each sentence into an equivalent sentence in disjunctive normal form. Simplify your answer as much as possible using the laws of associativity, commutativity, and idempotence. At each step in your chain, indicate which principle you are applying. Assume that A, B, C, and D are literals.

4.41  \( C \land (A \lor (B \land C)) \)  
4.42  \( B \land (A \land B \lor (A \lor B \land (B \land C))) \)  
4.43  \( A \land (A \land (B \lor (A \lor C))) \)