

## Chapter 3

# Semantic Reasoning Methods

### §3.1 Introduction

In this chapter, we present several basic methods for determining whether a given set of premises propositionally entails a given conclusion. In the next two chapters, we look at additional methods for making this determination.

### §3.2 Truth Table Method

One way of determining whether or not a set of premises logically entails a possible conclusion is to check the truth table for the logical constants of the language. This is called the *truth table method* and can be formalized as follows. Starting with a complete truth table for the propositional constants, iterate through all the premises of the problem, for each premise eliminating any row that does not satisfy the premise. Do the same for the conclusion. Finally, compare the two tables. If every row that remains in the premise table, i.e. is not eliminated, also remains in the conclusion table, i.e. is not eliminated, then the premises logically entail the conclusion.

As an example of this method, consider the problem from the introduction. We have two *premises*. If Mary loves Pat, then Mary loves Quincy. If it is Monday, then Mary loves Pat or Quincy. From these premises, we can draw the *conclusion* that on Mondays Mary loves Quincy. More formally, we can say that the premises logically entail the conclusion. To see this, let us set up the problem with three logical constants. The logical constant  $m$  means that it is Monday. The logical constant  $p$  means that Mary loves Pat. The logical constant  $q$  means that Mary loves Quincy.

Given this language, we can write our first premise as shown below. If Mary loves Pat, then Mary loves Quincy.

$$p \Rightarrow q$$

The following sentence captures the second premise. If it is Monday, then Mary loves Pat or Quincy.

$$m \Rightarrow p \vee q$$

Finally, our conclusion (which is to be proved) can be written as follows. If it is Monday, then Mary loves Quincy.

$$m \Rightarrow q$$

The following figure illustrates the handling of the premises. On the left is the full truth table. In the second table, we have processed the first premise by crossing out all rows in which  $p$  is true and  $q$  is false, as required by the semantics for the  $\Rightarrow$  operator. The table on the right shows the result after the second premise is taken into account as

well. We have crossed out the one additional row in which  $m$  is true and in which both  $p$  and  $q$  are false.

$m$	$p$	$q$	$m$	$p$	$q$	$m$	$p$	$q$
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>false</i>	×	×	×	×	×	×
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	×	×	×
<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>false</i>	×	×	×	×	×	×
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>

Now, we do the same for the conclusion. The tables below show the results. Again, the table on the left is the starting point. The table on the right is the result of crossing out any rows that do not satisfy the conclusion. We have eliminated any row in which  $m$  is true and  $q$  is false.

$m$	$p$	$q$	$m$	$p$	$q$
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>false</i>	×	×	×
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>false</i>	×	×	×
<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>

Finally, in order to make the determination of logical entailment, we compare the two rightmost tables and notice that every row remaining in the premise table also remains in the conclusion table. In other words, the premises logically entail the conclusion.

The truth table method has the merit that it is easy to understand. It is a direct implementation of the definition of logical entailment. In practice, it is awkward to manage two tables, especially since there are simpler approaches in which only one table needs to be manipulated.

### §3.3 Validity Checking

One of these approaches is the validity checking method. It goes as follows. To determine whether a set of sentences  $\{\varphi_1, \dots, \varphi_n\}$  logically entails a sentence  $\varphi$ , we form the sentence  $(\varphi_1 \wedge \dots \wedge \varphi_n \Rightarrow \varphi)$  and check that it is valid.

To see how this method works, consider the problem of Mary and Pat and Quincy once again. In this case, we write the tentative conclusion as shown below.

$$(p \Rightarrow q) \wedge (m \Rightarrow p \vee q) \Rightarrow (m \Rightarrow q)$$

We then form a truth table for our language with an added column for this sentence and check its satisfaction under each of the possible interpretations for our logical constants.

<i>m</i>	<i>p</i>	<i>q</i>	$(p \Rightarrow q) \wedge (m \Rightarrow p \vee q) \Rightarrow (m \Rightarrow q)$
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>

### §3.4 Unsatisfiability Checking

The *satisfiability checking method* is another single table approach. It is almost exactly the same as the validity checking method, except that it works negatively instead of positively. To determine whether a finite set of sentences  $\{\varphi_1, \dots, \varphi_n\}$  logically entails a sentence  $\varphi$ , we form the sentence  $(\varphi_1 \wedge \dots \wedge \varphi_n \wedge \neg \varphi)$  and check that it is unsatisfiable.

Both the validity checking method and the satisfiability checking method require about the same amount of work as the truth table method, but they have the merit of manipulating only table. As we shall see later, there are additional advantages when we move to more complex logics.

1.