Example 1.17 Consider now the formula $((A \to B) \to A) \wedge \neg A$. Suppose we want to determine whether this formula is satisfiable or not. Again we compute a truth table.

A	B	$(A \to B)$	$((A \to B) \to A)$	$\neg A$	$((A \to B) \to A) \wedge \neg A$
0	0	1	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	1	1	1	0	0

This formula is unsatisfiable. It is a contradiction.

Theoretically, we can determine whether any formula F is valid, satisfiable or unsatisfiable by looking at a truth table. Unfortunately, this is not always an efficient method. If F contains n atomic formulas, then there are 2^n rows to compute in the truth table for F. So if F happens to have, say, 23 atomic formulas, then computing a truth table is not feasible. One of our aims in this chapter is to find alternative methods for resolving the validity and satisfiability problems that avoid truth tables. More generally, our aim is to contrive various ways of determining whether or not a given formula is a consequence of a given set of formulas. This is a central problem of any logic.

1.3 Consequence and equivalence

We now introduce the fundamental notion of consequence. First, we define what it means for one formula to be a consequence of another. Later in this section, we similarly define what it means for a formula to be a consequence of a *set* of formulas.

Definition 1.18 Formula G is a *consequence* of formula F if for every assignment $\mathcal{A}$, if $\mathcal{A} \models F$ then $\mathcal{A} \models G$. We denote this by $F \models G$.

Note that the symbol $\models$ is used in a variety of ways. There is always a formula to the right of this symbol. When we write $\_\_ \models F$, the interpretation of "$\models$" depends on how we fill in the blank. The blank may either be filled with an assignment $\mathcal{A}$, a formula G, or not filled with the empty set. The three corresponding interpretations for $\models$ are as follows:

- $\mathcal{A} \models F$ means that $\mathcal{A}(F) = 1$. We read this as "$\mathcal{A}$ models F."
- $G \models F$ means every assignment that models G also models F. That is, F is a consequence of G.

- $\models F$ means every assignment models F. That is, F is a tautology.

So although $\models$ has multiple interpretations, in context it is not ambiguous.

The notion of consequence is closely related to the notion of "implies" discussed in Section 1.1. A formula G is a consequence of a formula F if and only if "F implies G" is always true. We restate this as the following proposition.

Proposition 1.19 For any formulas F and G, G is a consequence of F if and only if $F \to G$ is a tautology.

Proof We show that $F \to G$ is *not* a tautology if and only if G is *not* a consequence of F.

By the definition of "tautology," $F \to G$ is not a tautology if and only if there exists an assignment $\mathcal{A}$ such that $\mathcal{A} \models \neg(F \to G)$.

By the definition of "$\to$," $\mathcal{A} \models \neg(F \to G)$ if and only if $\mathcal{A} \models \neg(\neg F \vee G)$.

By the semantics of propositional logic, $\mathcal{A} \models \neg(\neg F \vee G)$ if and only if both $\mathcal{A} \models F$ and $\mathcal{A} \models \neg G$.

Finally, by the definition of "consequence," there exists an assignment $\mathcal{A}$ such that $\mathcal{A} \models F$ and $\mathcal{A} \models \neg G$ if and only if G is not a consequence of F. $\square$

Suppose we want to determine whether or not formula G is a consequence of a formula F. We refer to this as the *consequence problem*. By Proposition 1.19 this can be rephrased as a validity problem (since G is a consequence of F if and only if $G \to F$ is valid). Such problems can be resolved by computing a truth table. If the truth values for $F \to G$ are all 1s, then G is a consequence of F. Otherwise, it is not. In particular, if F is a contradiction, then G is a consequence of F regardless of G.

Example 1.20 Let F and G be formulas. Each of the following can easily be verified by computing a truth table.

$$(F \wedge G) \models F$$

$$F \models (F \vee G)$$

$$(F \wedge \neg F) \models G$$

Definition 1.21 If both G is a consequence of F and F is a consequence of G, then we say F and G are *equivalent*. We denote this by $F \equiv G$.

It follows from Proposition 1.19 that two formulas F and G are equivalent if and only if $F \leftrightarrow G$ is a tautology. So we can determine whether two formulas F and G are equivalent by computing a truth table. Each of the equivalences in the following examples can easily be verified in this manner.

Example 1.22 For all formulas F and G, $(F \wedge G) \equiv (G \wedge F)$ *and* $(F \vee G) \equiv (G \vee F)$.

Example 1.23 For any formula F and any tautology T, $(F \wedge T) \equiv F$ *and* $(F \vee T) \equiv T$.

Example 1.24 For any formula F and any contradiction $\perp$, $(F \wedge \perp) \equiv \perp$ *and* $(F \vee \perp) \equiv F$.

Example 1.25 (Distributivity rules) The following two equivalences exhibit the distributivity rules for $\wedge$ and $\vee$. For all formulas F, G, and H,

$$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H)) \text{ and}$$
$$(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H)).$$

Example 1.26 (DeMorgan's rules) For all formulas F and G

$$\neg(F \wedge G) \equiv (\neg F \vee \neg G), \text{ and}$$
$$\neg(F \vee G) \equiv (\neg F \wedge \neg G).$$

The equivalences in the previous examples are basic. Note that we refer to some of these equivalences as "rules." Each of these holds true for arbitrary formulas. From these basic equivalences, more elaborate equivalences can be created.

Example 1.27 Using the equivalences in the previous examples, we show that $((C \wedge D) \vee A) \wedge ((C \wedge D) \vee B) \wedge (E \vee \neg E) \equiv (A \wedge B) \vee (C \wedge D)$. Let L denote the formula on the left in this equivalence. Note that $(E \vee \neg E)$ is a tautology. By Example 1.23, L is equivalent to $((C \wedge D) \vee A) \wedge ((C \wedge D) \vee B)$. According to the second distributivity rule in Example 1.25, this is equivalent to $(C \wedge D) \vee (A \wedge B)$ (viewing $(C \wedge D)$ as the formula F in that rule).

By Example 1.22, this is equivalent to $(A \wedge B) \vee (C \wedge D)$ which is the formula on the right in our equivalence.

Using the basic rules in Examples 1.22–1.26, we were able to verify that $((C \wedge D) \vee A) \wedge ((C \wedge D) \vee B) \wedge (E \vee \neg E) \equiv (A \wedge B) \vee (C \wedge D)$. This is itself a rule, holding for any formulas A, B, C, D, and E. Alternatively, we could have verified this equivalence by computing a truth table. Such a truth table would have had $2^5 = 32$ rows. The previously established rules provided a more efficient method of verification.

Likewise, we could state "rules for consequence" that would allow us to show that one formula is a consequence of another without having to compute truth tables. In the next section, we exploit this idea and introduce the notion of formal proof. Formal proofs allow us to "derive" formulas from sets of formulas. The following definition extends the notion of consequence to this setting.

Definition 1.28 Let $\mathcal{F} = \{F_1, F_2, F_3, \ldots\}$ be a set of formulas.

For any assignment $\mathcal{A}$, we say $\mathcal{A}$ *models* $\mathcal{F}$, denoted $\mathcal{A} \models \mathcal{F}$ if $\mathcal{A} \models F_i$ for each formula F_i in $\mathcal{F}$.

We say a formula G is a *consequence* of $\mathcal{F}$, and write $\mathcal{F} \models G$, if $\mathcal{A} \models \mathcal{F}$ implies $\mathcal{A} \models G$ for every assignment $\mathcal{A}$.

Suppose that we want to determine whether a formula G is a consequence of a set of formulas $\mathcal{F}$. If $\mathcal{F}$ is finite, then we could consider the conjunction $\bigwedge \mathcal{F}$ of all formulas in $\mathcal{F}$ and compute a truth table for $\bigwedge \mathcal{F} \to G$. This method would certainly produce an answer. However, if the set $\mathcal{F}$ is large, then computing such a truth table is neither an efficient, nor a pleasant, thing to do. If $\mathcal{F}$ is infinite, then this method does not work at all. Another approach is to *derive* G from $\mathcal{F}$. Consider the following example.

Example 1.29 Let $\mathcal{F}$ be the following set of formulas

$$\{A, (A \to B), (B \to C), (C \to D), (D \to E), (E \to F), (F \to G)\}.$$

Suppose each of the seven formulas in $\mathcal{F}$ is true. Then, in particular, A and $A \to B$ are true. It follows that B must also be true. Likewise, since B and $B \to C$ are true, then C must also be true, and so forth. If each formula in $\mathcal{F}$ is true, then A, B, C, D, E, F, and G are true. Each of these formulas is a consequence of $\mathcal{F}$. We do not need a truth table to see this.

Let $\bigwedge \mathcal{F}$ be the conjunction of all formulas in $\mathcal{F}$. That is,

$$\bigwedge \mathcal{F} = A \wedge (A \to B) \wedge (B \to C) \wedge (C \to D) \wedge (D \to E) \wedge (E \to F) \wedge (F \to G).$$

The truth table for $\bigwedge \mathcal{F} \to G$ comprises 128 rows. Without computing a single row, we can see that each row will have truth value 1. The formula $\bigwedge \mathcal{F} \to G$ is a tautology and, equivalently, G is a consequence of $\mathcal{F}$.

In the previous example, we repeatedly used the fact that if X and $X \to Y$ are both true, then Y is also true. That is, we used the fact that Y is a consequence of $X \wedge (X \to Y)$. This follows from the truth table we computed in Example 1.16. Rather than compute another truth table (having 128 rows), we used a truth table we have already computed (having only four rows) to deduce that G is a consequence of $\mathcal{F}$. We *derived* G from $\mathcal{F}$ using a previously validated *rule*.

1.4 Formal proofs

A logic, by definition, has rules for deducing the truth of one sentence from that of another. These rules yield a system of formal proof. In this section, we describe such a proof system for propositional logic.