Chapter 3

⋆ Boolean Algebras and Circuits

There are 10 types of people in this world: those who understand binary numbers and those who don't.

- Anonymous.

At the end of the last chapter we noted a close analogy between Equivalence Laws for Propositional Logic on the one hand, and Set Identities on the other. In this chapter we explore this connection by looking at ***Boolean algebras***, the mathematical structures underlying both propositional logic and sets.

This analogy extends to the world of digital computers and other electronic devices, which are built from circuits which have binary inputs and outputs; that is, they manipulate values from the set $\mathbb{B} = \{0, 1\}$. At the implementation level these binary inputs and outputs are delivered by voltages on wires, with a low voltage being interpreted as 0 and a high voltage being interpreted as 1. The simplest components of digital circuits, ***logic gates***, are based on the connectives of propositional logic, with 0 (low voltage) and 1 (high voltage) being interpreted as **F** (false) and **T** (true), respectively. Composing logic gates together to create ever more complicated electronic components can thus be done in a way which is amenable to analysis via propositional logic. In this chapter we shall examine the fundamental role of Boolean algebra in underlying the building blocks of digital computers.

3.1 Boolean Algebras

A ***Boolean algebra*** is a set B which contains (at least) two distinct special elements 0 and 1, referred to as ***zero*** and ***unit***, respectively, along with two binary operators $+$ and $\cdot$, referred to as ***sum*** and ***product***, as well as a unary operator $'$, referred to as ***complementation***. That is, for every pair (x, y) of elements of B there are three further (but not necessarily different) elements of B denoted $x+y$, $x \cdot y$, and x'. These operators must all satisfy the ten ***Laws of Boolean Algebra*** given in Figure 3.1.

F. Moller, G. Struth, *Modelling Computing Systems*,
Undergraduate Topics in Computer Science,
DOI 10.1007/978-1-84800-322-4_4,