4. Boolean Algebra and Logical Design

"Contrariwise," continued Tweedledee, "if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic." Lewis Carroll, *Through the Looking Glass*.

Aristotelian logic

Before considering the work of George Boole and its application to the logical design of circuits used in computers, we shall have a brief look at a type of logical reasoning proposed by Aristotle in the fourth century B.C. The Aristotelian *syllogism* is a type of reasoning in which inferences are made from a pair of propositions termed *premisses* to a third proposition termed the *conclusion*. One commonly cited syllogism is the following:

All men are mortal.

Socrates is a man.

Therefore, Socrates is mortal.

This is an example of a valid syllogism since the conclusion follows from the two premisses. The truth or falsity of the premisses is another question. What is important is that if the premisses are true, then the conclusion derived from them is true.

As another example consider the following syllogism:

No mammals can fly.

All pigs are mammals.

Therefore, no pig can fly.   .

This is a valid syllogism since the conclusion follows from the two premisses even though the first premiss is false since bats, which are mammals, can fly. As a third example the syllogism

Fred is a person.

Some persons are very intelligent.

Therefore, Fred is very intelligent.

is not valid since the conclusion, even though it may be true, does not necessarily follow from the premisses.

All propositions occurring in a syllogism must be one of the four types which are given below together with their abbreviations and an example. (The upper-case letters in the abbreviations represent

subject and predicate, respectively, and the lower-case letters come from the vowels in the Latin words *affirmo*, I affirm, and *nego*, I deny):

All S are P	SaP	(All plants are vegetables.)
No S are P	SeP	(No plants are animals.)
Some S are P	SiP	(Some plants are fossils.)
Some S are not P	SoP	(Some plants are not fossils.)

Finally let M represent the middle term which occurs in each of the premisses of any syllogism but not in the conclusion. Then for example the first syllogism given above may be represented as MaP, SaM, SaP. The middle term may occur as the subject of the first premiss and the object in the second premiss as in the first example, or as the object in both premisses, or as the subject of both premisses, or as the object in the first and the subject in the second. These four categories may be represented by I, II, III and IV, respectively. Thus the first syllogism may be represented as Iaaa, the second as Ieae, and the third as IVaia..

In total there are 256 different types of syllogisms, and Aristotle and his followers determined that 19 of these were valid. (Later the number was reduced to 8 by the removal of redundancies,) A mnemonic for remembering the valid syllogisms, or moods, is said to have appeared first in the writings of Pope John XXI in the thirteenth century and began as follows:

Barbara, Celarent, Darii, Ferioque, prioris;
Cesare, Camestres, Festino, Baroko, secundae; …

Thus the first two syllogisms may be designated as "Barbara" and "Celarent", respectively, while the third does not appear in the list as it is not valid. In total there are 256 different types of syllogisms, and Aristotle and his followers determined that only nineteen of these were valid. Later the number was reduced to eight by the removal of redundancies

There are two principle objections to Aristotle's claim to the importance of this line of reasoning. First, it is obviously false that all arguments may be put in the form of a syllogism. For example, almost no mathematical arguments may be formulated in this way. Secondly, for those arguments which may be stated as syllogisms, the difficulty lies in posing the argument in this way. Once this is done, the determination of its validity is trivial. "In syllogistic inference," Bertrand Russell remarked, "you are suuposed to know already that all men are mortal and that Socrates is a man, hence you deduce what you never suspected before, that Socrates is mortal. This form of logic does actually occur, though very rarely."

Aristotelian logic dominated European thinking until the nineteenth century. It has been recorded that in the fourteenth century "Bachelors and Masters of Arts [at Oxford] who do not follow Aristotle's

philosophy are subject to a fine of 5p for each point of divergence." In 1797 Immanuel Kant said that logic was "a closed and completed body of doctrine".

In 1886 Lewis Carroll published *The Game of Logic* which was intended to introduce Aristotelian logic to young children. Included with the book was a playing board and a set of counters with examples for using this equipment for solving syllogisms. The English journalist and novelist Marghanita Laski who introduced this game to her children in connection with a friend's study of Carroll. She said that "after page thirteen there was no pretending that this was a game for children any more – at least, Mr. Carroll could go on pretending but I couldn't … I think – I hope – that the other seventy-five pages of the book were meant for post-graduate university students, for I gave up when my children did."

George Boole

George Boole was born in Lincoln, England in 1815. His father was a shoemaker who was more interested in science, especially the making of telescopes, than he was in making shoes. He made sure that his son received a good education, and from an early age George showed an aptitude for languages. By the age of 10 he was fluent in classical Latin and Greek, and later taught himself French, German and Italian.

When George was 16 his father's business collapsed, and he was forced to take up school teaching to support his parents and three younger siblings. At the age of 19 he opened his own school, and over the next dozen years or so he taught in various schools, some of which he founded himself, in Lincoln and the surrounding area.

His teaching duties did not prevent him from pursuing his mathematical interests and he soon was publishing original research in the *Cambridge Journal of Mathematics*. In 1844 Boole was awarded the first gold medal for mathematics by the Royal Society of London. Some of Boole's work at this time found application later in the development of Einstein's Special Theory of Relativity.

A deeply religious person, Boole considered the human mind to be of divine creation and turned his attention to how the mind processed information. He first published his revolutionary ideas in 1847 in *The Mathematical Analysis of Logic*, a book which although not widely read came to the attention of many of Britain's leading mathematicians and led to his first, and only, academic position.

In 1849 Boole was appointed the first professor of mathematics at the newly founded Queen's University Cork, now University College Cork, in spite of having neither a secondary education nor a university degree. In addition to conscientiously fulfilling his teaching duties, Boole further developed his ideas in mathematical logic, and in 1854 published his classic *An Investigation of the Laws of Thought*, praised by Bertrand Russell as "the work in which pure mathematics is discovered".

In 1855, at the age of nearly forty, Boole married Mary Everest, a girl half his age and the niece of Sir

George Everest after whom Mt. Everest is named. Their marriage was apparently a very happy one, and they had five daughters some of whom had distinguished careers in science as did some of their children.

Boole's career was cut short very tragically at the age of 49 when on a cold November day he walked two miles through the rain and lectured all day in wet clothes. When he returned home, his wife who believed that a remedy for an illness should resemble the cause put him to bed and doused him with buckets of cold water. Boole died of pneumonia on December 8, 1864. He was buried in a church cemetery in Blackrock, a suburb of Cork. Today the markings on the gravestone are almost illegible, but visitors may read a commemorative plaque inside the church.

During his lifetime Boole received many honorary degrees and was awarded many prizes. In 1857 he was elected a Fellow of the Royal Society. He is commemorated by the Boole Memorial Window in Queen‘s College erected in 1866, more recently by the Boole Library at University College, the annual Boole Memorial Lecture, and by the Boole Crater on the moon. His algebra of logic is referred to as Boolean algebra, which the *American Heritage Dictionary* defines as "Any of the algebraic systems based on mathematical forms and relationships borrowed from the symbolic logic of George Boole" who is described in the previous entry as "British mathematician, logician, and author."

Boolean algebra

Boole proposed that algebraic symbols could stand for classes of objects rather than just numbers so that there is an algebra of classes similar to the customary algebra of numbers. As a very simple example of Boole's algebra, let x represent the class of all cats and y the class of all orange things. Then xy, or $x \times y$, represents the class of all things which are cats and are orange, i.e., the class of all orange cats. Also the expression $x + y$ represents the class of all cats together with the class of all orange things, and so is a motley class which contains cats (all of them), golden retrievers, oranges, etc. The expression $x \times x$ which may be written as x^2 is the class of all cats which are cats which is simply the class of all cats. Therefore $x^2 = x$ for all values, i.e., classes, x, a relationship which is not generally true in conventional algebra. We shall define a Boolean algebra more formally in the following paragraph.

A Boolean algebra may be defined as a collection of objects $x, y, z, \ldots$, and the operations of union $+$, intersection $\times$, and complementation $'$ subject to the following fundamental postulates:

Commutation:	$x + y = y + x$
	$x \times y = y \times x$
Distribution:	$x + (y \times z) = (x + y) \times (x + z)$
	$x \times (y + z) = (x \times y) + (x \times z)$
Identity:	$x + \boldsymbol{0} = x$
	$x \times \boldsymbol{1} = x$

Complementation: $x + x' = \boldsymbol{1}$

$x \times x' = \boldsymbol{0}$

The objects $\boldsymbol{0}$ and $\boldsymbol{1}$ are known as identity elements, and the equality sign = indicates that the two arguments represent identical collections of objects.

These postulates all have a reasonable interpretation. For example the first commutation postulate may be interpreted as "all things that have the property x or the property y are identical to all things that have the property y or the property x". Also the first complementation law may be interpreted as "all things which have the property x or do not have the property x is identical to the universe of all objects under consideration".

Of the many theorems that may be deduced from these postulates we shall mention only De Morgan's Laws $(x \times y)' = x' + y'$ and $(x + y) = x' \times y'$. Also we may note that the inclusion of one class of objects in another may be expressed by the statement $x \times y = x$ which implies that x is contained in y. This inclusion may also be expressed as $x \times y' = \boldsymbol{0}$.

Now let us consider how Aristotelian syllogisms may be expressed and proven, or disproven, using Boolean algebra, using as an example the familiar one given at the beginning of the chapter:.

All men are mortal.

Socrates is a man.

Therefore, Socrates is mortal.

If x represents the class of all men, y the class of all mortal beings and z the class of Socrates, then the two premisses are $x \times y = x$ and $z \times x = \mathrm{x}$ and the conclusion $z \times y = z$. The proof is given as follows:

$x \times y = x$	First premiss
$(x \times y)' = x'$	Complementation
$x' + y' = x'$	De Morgan's Law
$z \times (x' + y') = z \times \mathrm{x}'$	
$(z \times x') + (z \times y') = z \times x'$	
$\boldsymbol{0} + (z \times y') = \boldsymbol{0}$	
$z \times y' = \boldsymbol{0}$	Identity

In words, the class consisting of Socrates which are not in the class of mortal beings is empty.

As an example of how Boolean expressions may be simplified consider the expression

$(x' + y') \times (x + y') \times (x + y)$

where we shall let x represent, as before, the class of all cats and y the class of all orange cats. The simplification is as follows:

$(x' + y') \times (x + y') \times (x + y)$

$(x' + y') \times (x \times (y' + y))$ Distribution

$(x' + y') \times x \times \mathbf{\mathit{1}}$	Complementation
$(x' + y') \times x$	Identity
$(x' \times x) + (y' \times x)$	Distribution
$\mathit{0} + (y' \times x)$	Complementation
$y' \times x$	Identity
$x \times y'$	Commutation

This last expression may be interpreted simply as "the class of all cats which are not orange".

In addition to interpreting Boolean algebra in terms of the algebra of sets as we have done above, Boolean algebra may also be interpreted as Boole did in terms of the calculus of propositions. Instead of sets, the basic elements may be considered to be propositions p, q, r, ... which may be either true **T** or false **F**. An expression $p \vee q$ would be intrepreted as "either p is true or q is true or both are true" and the commutative postulate $p \vee q = q \vee p$ would have have a meaningful interpretation. In 1937 Claude Shannon, then a graduate student at the Massachusetts Institute of Technology, completed what has been termed the "most important master's thesis ever written" in which he showed the correspondence between the calculus of propositions and electrical circuits, therby making the results of the calculus applicable to the design and analysis of electrical circuits. For example the variables X, Y, Z, ... could represent electrical switches. If switch X is closed, the variable X will have the value 1; if switch X is open, variable X will have the value 0. A circuit consisting of two switches X and Y in parallel will be represented by X ∨ Y, and a circuit consisting of two switches in series will be represented by X ∧ Y. Thus the postulates and theorems of the calculus of propositions are available for use in the design and simplification of electrical circuits. We shall limit our remarks in the following section to the use of Boolean algebra for the logical design of simple circuits for the addition of binary numbers.

Truth tables

A logical function, i.e., a function with logical or Boolean arguments limited to *0* and *1* and which gives Boolean results, may be conveniently defined by a *truth table* which one author has defined as "a conspectus of the various possibilities that may arise". In this section we shall introduce a few additional **J** primitive verbs which will be required in the following section on binary addition. For convenience we shall define the two-item boolean list `B=: 0 1`.

The monadic verb *not* `-.` gives the complement of its Boolean argument, and `-. 0` is *1* and `-. 1` is *0*. The truth table is given by `B ,. -.B` which is a two-column table with `B` in the first column and the corresponding values of `-.B` in the second:

```
0 1
1 0
```

The following dialogue will give truth tables for the dyadic Boolean verbs *or* `+.`, *and* `*.` and *not-equal* `~:`:

```
   B by B over B +./ B
+-+---+
| |0 1|
+-+---+
|0|0 1|
|1|1 1|
+-+---+
   B by B over B *./ B
+-+---+
| |0 1|
+-+---+
|0|0 0|
|1|0 1|
+-+---+
   B by B over B ~:/ B
+-+---+
| |0 1|
+-+---+
|0|0 1|
|1|1 0|
+-+---+
```

Binary addition

The addition of two digits in a number system to any base gives a two-digit result consisting of a sum digit and a carry digit, either or both of which may be zero. For example, in the decimal system the expression 6 + 9, say, which is equal to 15 gives a sum digit of 5 and a carry digit of 1, and 3 + 4 is 7 with a carry digit of 0. The binary system is especially simple as the digits being added and the sum and carry digits are limited to the values 0 and 1. In this section we shall consider the logical design of adders which will find the sum of two binary digits, the sum of two binary digits and a carry digit, and finally the sum of two arbitrary binary numbers of equal length.

A binary adder, which gives the sum and carry digits resulting from the addition of two binary digits, is termed a two-input adder or a half-adder, and may be defined by the following truth table:

x	y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0

```
 1  1   0    1
```

The sum digit is `1` if and only if the two digits `x` and `y` being summed are unequal, and the carry digit is `1` if and only if both digits are equal to `1`. Thus the sum and carry digits may be given by the **J** expressions `~:/x,y` and `*./x,y`, respectively, and we may define the following verb for a half-adder:

```
HalfAdder=: ~:/,*./ .
```

A full-adder, also called a three-input adder, gives the sum and carry digits produced by the sum of two binary digits and a carry digit resulting from the addition of two previous binary digits. If x and y represent the two binary digits and z the carry digit, then the truth table for a full-addrer is given by the following table:

x	y	z	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The following simple example shows the addition of the two binary numbers 1 1 0 1 1 1 1 and 0 1 1 0 1 0:

1 1 1 1 0 0	Carry from previous column	
1 1 0 1 1 1	First number x	55 (base 10)
0 1 1 0 1 0	Second number y	26
1 0 1 0 0 0 1	Sum	81

A careful examination of the above truth table and the example should help suggest the following method of implementing a full-adder from two half-adders:

- Use a first half-adder to find the sum digit s_1 and the carry digit c_1 resulting from the addition of x and y.
- Use a second half-adder to find the sum digit s_2 and the carry digit c_2 resulting from the sum of s_1 and the previous carry z.
- The output of the full-adder is the sum digit s_2 and the carry digit which is 1 if either carry digit c_1 or c_2 is 1.

A **J** verb for a full-adder is as follows:

```
FullAdder=: 3 : 0
:
```

```
z=.x.['x y'=. y.
'sl c1'=. HalfAdder x,y
's2 c2'=. HalfAdder s1,z
s2, (c1=1)+.(c2=1)
)
```

A full- adder may be used in the design of an adder which will find the sum of two arbitrary binary numbers. The following explicit function `BinaryPlus` gives the sum of its two arguments assumed to be of equal length and is followed by some examples of its use:

```
BinaryPlus=: 3 : 0
:
'X Y'=. x.;y.
R=. i.0
if. (#X) ~: #Y do.
   return.
else.
   Z=. 0
 while.  0<#X do.
    SumCarry=. Z FullAdder ({:X),{:Y
    R=. ({.SumCarry),R
    X=. }:X
    Y=. }:Y
    Z=. {:SumCarry
 end.
end.
if. Z = 1 do. R=. 1,R end.
R
)
0 0 1 BinaryPlus 0 1 1
1 0 0
   1 0 1 1 1 BinaryPlus 0 0 1 0 1
1 1 1 0 0
   1 0 1 0 BinaryPlus 1 1 0

   1 1 0 1 1 1 BinaryPlus 0 1 1 0 1 1
```

```
1 0 1 0 0 1 0
   1 1 BinaryPlus 1 1
1 1 0
```