Mathematics for Computer Scientists

Lecture notes for the module G51MCS

Venanzio Capretta
University of Nottingham
School of Computer Science

Chapter 1

Boolean Algebra

Ordinary algebra is that area of mathematics concerned with formulas denoting numbers. It deals with formulas like $a + (-b \times c)$. It uses letters like a, b, c to indicate *variables* that represent unknown numbers. Then it combines them using operations like addition, $+$, multiplication, $\times$, opposite, $-$.

Boolean algebra is similar, but instead of numbers it is concerned with truth values, also called *Boolean values*, true and false. It deals with propositional formulas like $A \vee (\neg B \wedge C)$. It uses letters like A, B, C to indicate variables representing unknown truth values. Then it combines them using connectives like disjunction, $\vee$, conjunction, $\wedge$, negation, $\neg$, and implication, $\Rightarrow$.

1.1 Intuitionistic and Classical Logic

The rules that we have seen so far constitute what is called *intuitionistic logic*. Traditionally another rule is added to the system to give what is called *classical logic*:

m	$\neg\neg A$	
$\vdots$	$\vdots$	
p	A	$\neg\neg$E, m

This rule, called *double negation elimination*, states that if we proved that it is impossible that A is false, then A must be true. One of its consequences is the *law of excluded middle*, that states that every proposition is either true or false:

1	$\neg(A \lor \neg A)$	
2	A	
3	$A \lor \neg A$	$\lor$I, 2
4	$\bot$	$\neg$E, 1, 3
5	$\neg A$	$\neg$I, 2–4
6	$A \lor \neg A$	$\lor$I, 5
7	$\bot$	$\neg$E, 1, 6
8	$\neg\neg(A \lor \neg A)$	$\neg$I, 1–7
9	$A \lor \neg A$	$\neg\neg$E, 8

Double negation elimination and the principle of excluded middle seem reasonable rules, but they have one undesired effect on our logic: they make it *non constructive*. Suppose A is a proposition that states that there is a solution to some problem. An intuitionistic proof of A would in that case give us a way to compute the solution. On the other hand, a classical proof of it would only tell us that it is impossible that the solution doesn't exist, without helping us to find it. Similarly, an intuitionistic proof of $A \lor B$ gives us a method to determine which of the two alternatives is valid, while a classical proof would only tell us that they cannot be both false.

For this reason intuitionistic logic is particularly interesting in computer science, where we care about being able to compute solutions to problems rather than just have an abstract guarantee that it is impossible that the solution doesn't exist.

From now on we make free use of double negation elimination, but always remember that an intuitionistic proof gives us more information than a classical one.

Even if the principle of excluded middle is a derived proposition, it is often useful to apply it in one step as if it were a rule on its own. After all, once we know that it holds, we should be able to use it freely:

$\vdots$	$\vdots$	
p	$A \lor \neg A$	EM

1.2 Examples of Classical Derivations

Here is a couple of examples of derivations that use double negation elimination or the principle of excluded middle in an essential way.

m | A n | B p | $A \wedge B$ $\wedge$I, m, n	m | $A \wedge B$ p | A $\wedge$E, m m | $A \wedge B$ p | B $\wedge$E, m
m | A p | $A \vee B$ $\vee$I, m m | B p | $A \vee B$ $\vee$I, m	m | $A \vee B$ h | | A i | | C j | | B k | | C p | C $\vee$E, m, h–i, j–k
m | | A n | | B $n+1$ | $A \Rightarrow B$ $\Rightarrow$I, m–n	m | $A \Rightarrow B$ n | A p | B $\Rightarrow$E, m, n
m | | A n | | $\bot$ $n+1$ | $\neg A$ $\neg$I, m–n	m | $\neg A$ n | A p | $\bot$ $\neg$E, m, n
m | $\bot$ p | C $\bot$E, m	m | A p | A R, m
m | $\neg\neg A$ p | A $\neg\neg$E, m	p | $A \vee \neg A$ EM

Figure 1.1: Natural Deduction Rules for Propositional Logic

1	$\neg(A \wedge B)$	
2	$A \vee \neg A$	EM
3	$\vert\ A$	
4	$\vert\ \vert\ B$	
5	$\vert\ \vert\ A \wedge B$	$\wedge$I, 3, 4
6	$\vert\ \vert\ \bot$	$\neg$E, 1, 5
7	$\vert\ \neg B$	$\neg$I, 4–6
8	$\vert\ \neg A \vee \neg B$	$\vee$I, 7
9	$\vert\ \neg A$	
10	$\vert\ \neg A \vee \neg B$	$\vee$I, 9
11	$\neg A \vee \neg B$	$\vee$E, 2, 3–8, 9–10

This principle, that we can derive $\neg A \vee \neg B$ from $\neg(A \wedge B)$ is one direction of one of the two de Morgan laws; we'll talk about them later. The vice versa, proving $\neg(A \wedge B)$ from $\neg A \vee \neg B$ is also valid and can be done with intuitionistic logic. I suggest you try.

The next example is also a famous one: it's known as Peirce's law.

1	$(A \Rightarrow B) \Rightarrow A$	
2	$\vert\ \neg A$	
3	$\vert\ \vert\ A$	
4	$\vert\ \vert\ \bot$	$\neg$E, 2, 3
5	$\vert\ \vert\ B$	$\bot$E, 4
6	$\vert\ A \Rightarrow B$	$\Rightarrow$I, 3–5
7	$\vert\ A$	$\Rightarrow$E, 1, 6
8	$\vert\ \bot$	$\neg$E, 2, 7
9	$\neg\neg A$	$\neg$I, 2–8
10	A	$\neg\neg$E, 9

To summarise all that we have learnt so far, Figure 1.1 gives all the rules of propositional logic in one table in a synthetic form.

1.3 Truth Tables

A way of determining the truth or falsity of a formula is to use *truth tables*. It is very different from the method of logical derivations, but equivalent. If we adopt the classical concept of logic, every proposition is either true or false (by the principle of excluded middle). This means that we can assign to it a *truth value*, either true or false. A complex proposition obtained by joining two simpler ones by a connective will have a truth value that depends in a predictable way from the truth values of its components. For example, the conjunction $A \wedge B$ will have the truth value true in the case that both A and B have the value true, it will have the truth value false in all other cases. Therefore, we can make the following table:

A	B	$A \wedge B$
true	true	true
true	false	false
false	true	false
false	false	false

The other connectives can be described similarly by truth tables:

A	$\neg A$
true	false
false	true

A	B	$A \vee B$
true	true	true
true	false	true
false	true	true
false	false	false

A	B	$A \Rightarrow B$
true	true	true
true	false	false
false	true	true
false	false	true

For more complex propositional formulas, we can construct truth tables by assigning all possible truth values to the atomic propositions in them and then compute recursively the values of the subformulas from the smaller ones to the largest one using the truth tables of the connectives:

A	B	$\neg B$	$A \vee \neg B$	$\neg A$	$A \vee \neg B \Rightarrow \neg A$
true	true	false	true	false	false
true	false	true	true	false	false
false	true	false	false	true	true
false	false	true	true	true	true

A formula whose truth table has always the value true for all possible values of its atomic propositions is called a *tautology*. This system of determining the truth of a proposition is equivalent to the propositional calculus that we studied before: a formula is a tautology exactly in the case that it can be derived in propositional logic without any hypothesis.

Using truth tables we can see that some connectives can be defined in terms of others. For example, implication can be defined in terms of negation and disjunction: $A \Rightarrow B$ is equivalent to $\neg A \vee B$, in the sense that their truth tables give the same results.

A	B	$A \Rightarrow B$
true	true	true
true	false	false
false	true	true
false	false	true

A	B	$\neg A$	$\neg A \vee B$
true	true	false	true
true	false	false	false
false	true	true	true
false	false	true	true

Conjunction can also be defined in terms of negation and disjunction: $A \wedge B$ is equivalent to $\neg(\neg A \vee \neg B)$. Verify it yourself by writing down their truth tables. In conclusion, we only need to have negation and disjunction in our set of basic Boolean operators. The rest can be constructed from them.

We can do even better. We can find a single connective from which all the other ones can be derived. There are many different ways of doing this. One is to use the *nand* operator (also called *Sheffer stroke*), which represents the negation of the conjunction. We use the symbol $\uparrow$ for it. Here is its truth table.

A	B	$A \uparrow B$
true	true	false
true	false	true
false	true	true
false	false	true

It's easy to check that $\neg A$ is equivalent to $A \uparrow A$ and $A \vee B$ is equivalent to $(\neg A) \uparrow (\neg B)$. Since we already know that the other operators can be realised by negation and disjunction, we can do everything with nand.

How many possible binary Boolean operators are there? We just have to count the number of different ways to construct a truth table for two variables A and B. Each table has four lines corresponding to the possible truth values for A and B. For each line we can choose whether our operator will return true or false. So we have two choices for each of the four lines: the total number of possible choices is then $2 \times 2 \times 2 \times 2 = 16$. There are sixteen possible binary operators.

But we don't need to worry about them: we can realise each of them using the ones that we already have. In fact we can realise every possible truth table by using only the nand operator. It is an instructive exercise to try this with a few randomly chosen tables. Take the following, for example.

A	B	?
true	true	true
true	false	false
false	true	false
false	false	true

You can verify that it is equivalent to $(A \wedge B) \vee (\neg A \wedge \neg B)$.

The general algorithm to associate a formula to an arbitrary truth table is this:

- Select all the rows for which the table gives true;
- For each of these rows, look at the values assigned to the variables;
- If the value is true take the variable by itself, if it's false take its negation;
- Make the conjunction of all the variables and variable negations that you wrote;
- Take the disjunction of the formulas that you obtained for each row.

This algorithm works also for more than two variables. Let's illustrate it with a three variables truth table.

A	B	C	?
true	true	true	true
true	true	false	true
true	false	true	false
true	false	false	false
false	true	true	false
false	true	false	true
false	false	true	false
false	false	false	true

There are four lines for which the table gives true, the first, second, sixth and eighth. The first line corresponds to the value true for all three variables, so we associate the formula $A \wedge B \wedge C$ to it. The second line corresponds to the value true for A and B and false for C, so we associate the formula $A \wedge B \wedge \neg C$ to it. The sixth line corresponds to the value true for B and false for A and C, so we associate the formula $\neg A \wedge B \wedge \neg C$ to it. The eighth line corresponds to the value false for all three variables, so we associate the formula $\neg A \wedge \neg B \wedge \neg C$ to it. Finally, we put the three formulas associated to the three true rows together using disjunction:

$$(A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge \neg C).$$

1.4 Logical Equivalences

A special kind of tautology is the logical equivalence. Given two propositional formulas A and B, we say that they are equivalent if their truth tables give exactly the same values for all possible assignments to the atomic formulas.

If you look at the example above, you will notice that the formulas $\neg A$ and $A \vee \neg B \Rightarrow \neg A$ always have the same truth value, so they are equivalent. We express this by writing $(A \vee \neg B \Rightarrow \neg A) \Leftrightarrow (\neg A)$ The equivalence symbol $\Leftrightarrow$ is a new connective, meaning implication in both directions. That is, $A \Leftrightarrow B$ means $(A \Rightarrow B) \wedge (B \Rightarrow A)$. As a connective, $\Leftrightarrow$ is given the lowest priority: all other connectives have precedence over it.

We defined equivalence in terms of truth tables. We could as easily define it using derivations: A is equivalent to B if there is one derivation with hypothesis A and conclusion B and another derivation with hypothesis B and conclusion A.

Many important logical rules can be expressed as equivalences. We saw the proofs of some of them (at least in one of the two directions) already. You can convince yourselves that they are true by computing their truth tables or by constructing derivations for them.

$A \wedge B \Leftrightarrow B \wedge A$	Commutativity of conjunction
$A \vee B \Leftrightarrow B \vee A$	Commutativity of disjunction
$A \wedge (B \wedge C) \Leftrightarrow (A \wedge B) \wedge C$	Associativity of conjunction
$A \vee (B \vee C) \Leftrightarrow (A \vee B) \vee C$	Associativity of disjunction
$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$	Distributivity of conj. over disj.
$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$	Distributivity of disj. over conj.
$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$	First De Morgan law
$\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$	Second De Morgan law

1.5 The laws of Boolean algebra

In ordinary algebra there is a calculus with precise laws to manipulate formulas and simplify them. Given a complex formula, we can use these laws to reduce it to a simpler one, like in this example:

$$\begin{aligned}
(a+b) \times c - b \times c &= a \times c + b \times c - b \times c && \text{distributivity of } \times \text{ over } + \\
&= a \times c + 0 && \text{defining property of opposite} \\
&= a \times c && \text{zero is the unit of addition}
\end{aligned}$$

Similarly in Boolean algebra we have laws to manipulate and simplify propositional formulas. Here is an example of the simplification of a formula in Boolean algebra (don't worry too much about the justifications on the right for the moment):

$$\begin{aligned}
A \wedge (C \vee \neg A) &= (A \wedge C) \vee (A \wedge \neg A) && \text{distributivity of } \wedge \text{ over } \vee \\
&= (A \wedge C) \vee \mathsf{false} && \text{contradiction} \\
&= A \wedge C && \mathsf{false} \text{ is the unit of disjunction}
\end{aligned}$$

As you can see, Boolean algebra is very similar to ordinary algebra, but not exactly the same. The first step we must take is to write down all the rules that are satisfied by the propositional connectives. Here they are.

The laws of Boolean algebra

$A \wedge B = B \wedge A$	Commutativity of conjunction
$A \vee B = B \vee A$	Commutativity of disjunction
$A \wedge (B \wedge C) = (A \wedge B) \wedge C$	Associativity of conjunction
$A \vee (B \vee C) = (A \vee B) \vee C$	Associativity of disjunction
$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$	Distributivity of conj. over disj.
$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$	Distributivity of disj. over conj.
$\neg(A \wedge B) = \neg A \vee \neg B$	First De Morgan law
$\neg(A \vee B) = \neg A \wedge \neg B$	Second De Morgan law
$A \wedge \mathsf{true} = A$	Unit of conjunction
$A \vee \mathsf{false} = A$	Unit of disjunction
$A \wedge \mathsf{false} = \mathsf{false}$	Zero of conjunction
$A \vee \mathsf{true} = \mathsf{true}$	Zero of disjunction
$A \wedge A = A$	Idempotence of conjunction
$A \vee A = A$	Idempotence of disjunction
$A \wedge (A \vee B) = A$	First absorption law
$A \vee (A \wedge B) = A$	Second absorption law
$A \wedge \neg A = \mathsf{false}$	Contradiction
$\neg\neg A = A$	Double negation
$A \vee \neg A = \mathsf{true}$	Excluded middle
$A \Rightarrow B = \neg A \vee B$	Definition of implication

We gave a single law for the implication connective. It allows us to rewrite a formula containing $\Rightarrow$ to one containing $\neg$ and $\vee$. We can use this *definition of implication in terms of negation and disjunction* to derive some extra laws that are useful in manipulating Boolean formulas containing implications.

Laws of Implication

$A \wedge (A \Rightarrow B) = A \wedge B$	Modus ponens
$A \Rightarrow B = \neg B \Rightarrow \neg A$	Contrapositive
$(A \wedge B) \Rightarrow C = A \Rightarrow (B \Rightarrow C)$	Currying

and many others.

As an example, here is how the rule of modus ponens can be derived:

$$\begin{aligned}
A \wedge (A \Rightarrow B) &= A \wedge (\neg A \vee B) && \text{Definition of implication}\\
&= (A \wedge \neg A) \vee (A \wedge B) && \text{Distributivity of conj. over disj.}\\
&= \mathsf{false} \vee (A \wedge B) && \text{Contradiction}\\
&= (A \wedge B) \vee \mathsf{false} && \text{Commutativity of disjunction}\\
&= A \wedge B && \text{Unit of disjunction}
\end{aligned}$$

The implication connective is not the only one that can be defined in terms of the others. We can also define conjunction in terms of negation and disjunction, like this:

$$A \wedge B = \neg(\neg A \vee \neg B).$$

Then we would need only the laws for $\neg$ and $\vee$ and those for $\wedge$ could be derived.

1.6 Proving the laws

The laws that we gave in the previous section need to be justified: how do we know that they are valid? We know two ways of doing this: with derivations or with truth tables.

Let's take, as an example, the first De Morgan law:

$$\neg(A \wedge B) = \neg A \vee \neg B.$$

For illustration, let me show you how we can prove it using both methods.

Proof with derivations We use propositional logic to prove the equality. We have to show that the two propositional formulas $\neg(A \wedge B)$ and $\neg A \vee \neg B$ are logically equivalent. To do this we need two derivations: the first showing that if we take $\neg(A \wedge B)$ as hypothesis, then we can derive $\neg A \vee \neg B$; the second showing that if we take $\neg A \vee \neg B$ as hypothesis, then we can derive $\neg(A \wedge B)$.

Here are both derivations:

1	$\neg(A \wedge B)$	
2	$A \vee \neg A$	EM
3	$\quad A$	
4	$\quad\quad B$	
5	$\quad\quad A \wedge B$	$\wedge$I, 3, 4
6	$\quad\quad \bot$	$\neg$E, 1, 5
7	$\quad \neg B$	$\neg$I, 4–6
8	$\quad \neg A \vee \neg B$	$\vee$I, 7
9	$\quad \neg A$	
10	$\quad \neg A \vee \neg B$	$\vee$I, 9
11	$\neg A \vee \neg B$	$\vee$E, 2, 3–8, 9–10

1	$\neg A \vee \neg B$	
2	$\quad A \wedge B$	
3	$\quad A$	$\wedge$E, 2
4	$\quad B$	$\wedge$E, 2
5	$\quad\quad \neg A$	
6	$\quad\quad \bot$	$\neg$E, 5, 3
7	$\quad\quad \neg B$	
8	$\quad\quad \bot$	$\neg$E, 7, 4
9	$\quad \bot$	$\vee$E, 1, 5–6, 7–8
10	$\neg(A \wedge B)$	$\neg$I, 2–9

Proof with truth tables To prove the law, we just have to make the truth tables of both propositional formulas and check that they are equal: they always give the same result for all possible values of A and B.

A	B	$A \wedge B$	$\neg(A \wedge B)$
true	true	true	false
true	false	false	true
false	true	false	true
false	false	false	true

A	B	$\neg A$	$\neg B$	$\neg A \vee \neg B$
true	true	false	false	false
true	false	false	true	true
false	true	true	false	true
false	false	true	true	true

Since both truth tables give the same result on every row, we can conclude that the two formulas are logically equivalent.

Now try for yourself: for each of the laws of Boolean algebra, prove its validity both using propositional logic and truth tables. With the first method: make two derivations showing that if you take one of the two equated formulas as hypothesis, you can derive the other. With the second method: make the truth tables for the two formulas and verify that they give the same result on every row.

1.7 Logical equivalence as a connective

Logical equivalence can be used as a new connective: two formulas A and B can be said to be equivalent if A implies B and also B implies A. We write $A \Leftrightarrow B$ in this case (read it *"A equivales B"*). So we define the new connective $\Leftrightarrow$ so:

$$A \Leftrightarrow B = (A \Rightarrow B) \wedge (B \Rightarrow A). \qquad \text{Definition of logical equivalence}$$

What is the difference between writing $A = B$ and $A \Leftrightarrow B$? They both have the same meaning: the two formulas A and B are logically equivalent. But we use them in different ways. When we write $A = B$ we mean it as an assertion: we are stating that the two formulas A and B are equivalent. When we write $A \Leftrightarrow B$ we are just denoting a new propositional formula obtained by applying the *equivale* connective to A and B; we are not declaring that the equivalence is true.

We give $\Leftrightarrow$ the lowest priority as a connective, so all other connectives must be parenthesised before it. Like the other connectives, $\Leftrightarrow$ has some laws that allow us to manipulate it in Boolean algebra.

Laws of equivalence

$$(A \Leftrightarrow B) \Leftrightarrow C = A \Leftrightarrow (B \Leftrightarrow C) \qquad \text{Associativity of equivalence}$$
$$A \Leftrightarrow B = B \Leftrightarrow A \qquad \text{Commutativity of equivalence}$$
$$A \Leftrightarrow B = \neg A \Leftrightarrow \neg B \qquad \text{Contrapositive of equivalence}$$

See if you can derive these laws in Boolean algebra using the definition of $\Leftrightarrow$ in terms of $\wedge$ and $\Rightarrow$ and their Boolean laws.

1.8 The laws of equality

We already silently used some laws of equality when we gave examples of the manipulation of formulas in Boolean algebra. We all instinctively know them, but it is useful to spell them out formally.

$A = A$	Reflexivity of equality
$A = B \Rightarrow B = A$	Symmetry of equality
$A = B \wedge B = C \Rightarrow A = C$	Transitivity of equality

Finally there is the *substitutivity of equality* or *congruence* law, that states that whenever two formulas are equal, we can replace one with the other:

$$A = B \Rightarrow F[A] = F[B].$$

Here F denotes any large formula that contains in it an occurrence of A or B. The law states that inside F we can replace A with B.

When we write a sequence of equalities in which every step is justified by one of the laws of Boolean algebra, for example:

$$F_1 = F_2 = F_3 = \cdots = F_{10},$$

we are implicitly using the law of congruence and that of transitivity: in each step we replace some part of the formula by something that is equivalent to it according to one of the laws; we conclude that the first formula F_1 is equivalent to the last F_{10} by the repeated application of transitivity to all the intermediate steps.

For example, consider the following proof of a Boolean equality:

$$\begin{aligned} A \vee (B \wedge \neg A) &= (A \vee B) \wedge (A \vee \neg A) && \text{distributivity of disj. over conj.} \\ &= (A \vee B) \wedge \mathsf{true} && \text{excluded middle} \\ &= A \vee B && \text{unit of conjunction} \end{aligned}$$

The second line is actually a combination of three steps:

1. $A \vee \neg A = \mathsf{true}$ by the law of excluded middle;

2. By the law of congruence we can replace $A \vee \neg A$ with true inside any formula, so $(A \vee B) \wedge (A \vee \neg A) = (A \vee B) \wedge \mathsf{true}$;

3. By the first line we have that $A \vee (B \wedge \neg A) = (A \vee B) \wedge (A \vee \neg A)$ and we just proved that $(A \vee B) \wedge (A \vee \neg A) = (A \vee B) \wedge \mathsf{true}$, so by transitivity $A \vee (B \wedge \neg A) = (A \vee B) \wedge \mathsf{true}$.

A similar sequence of steps is implicit in the third line.

1.9 Examples of derivation of Boolean equalities

To illustrate how we use the laws of Boolean algebra to prove equalities, let's start with a simple one. The law of distributivity of disjunction over conjunction tells us what we can do when we have a disjunction on the left of a conjunction. We would like to have a similar equality when the disjunction is on the right:

$$(A \wedge B) \vee C = (A \vee C) \wedge (B \vee C).$$

Let's prove it:

$$\begin{aligned} (A \wedge B) \vee C &= C \vee (A \wedge B) && \text{commutativity of disjunction} \\ &= (C \vee A) \wedge (C \vee B) && \text{distributivity of disj. over conj.} \\ &= (A \vee C) \wedge (B \vee C) && \text{commutativity of disjunction (twice)} \end{aligned}$$

To check that you understood, I suggest that you give yourself a proof of the right-side distributivity of conjunction over disjunction:

$$(A \vee B) \wedge C = (A \wedge C) \vee (B \wedge C).$$

For another example, let's prove this equality:

$$A \vee \neg(A \Rightarrow B) = A.$$

We start by rewriting the implication in terms of negation and disjunction and then we simplify using the laws of Boolean algebra:

$$\begin{aligned} A \vee \neg(A \Rightarrow B) &= A \vee \neg(\neg A \vee B) && \text{definition of implication} \\ &= A \vee (\neg\neg A \wedge \neg B) && \text{De Morgan} \\ &= A \vee (A \wedge \neg B) && \text{double negation} \\ &= (A \vee A) \wedge (A \vee \neg B) && \text{distributivity of disj. over conj.} \\ &= A \wedge (A \vee \neg B) && \text{idempotence of disjunction} \\ &= A && \text{absorption} \end{aligned}$$