CS304 – CS310
Mathematical and Logical Aspects of Computing

Lecture 3: Logical Equivalence, and the $\rightarrow$ operator

Tuesday, 10^{th} Sep 2013

Boolean Operators...

take arguments of *Boolean type* (i.e., with one of the true values F or T assigned to them) and maps these to one of F or T.

Important operators we've seen so far include:

- *Negation*, written as $\neg a$, read as "not a" (sometimes : $\sim a$ or $\overline{a}$).
- *Conjunction*, written as $a \wedge b$, read as "and" (sometimes : $a\&b$).
- *Disjunction*, written as $a \vee b$, read as "or".
- *Non-equivalence*, a.k.a., the *exclusive or* operator, and denoted as $\oplus$ (sometimes: $\not\equiv$ or `XOR`)

. . . . . .

In a few minutes we meet some other *VERY* important ones:

1. *nor*, written as $a \downarrow b$; (sometimes called the "Peirce arrow").
2. *nand*, written as $a \uparrow b$;
3. *equivalence*, written as $a \leftrightarrow b$.
4. *implication*, written as $a \rightarrow b$, and read as "a implies b";

Assignment: Giving a particular value or F or T to a Boolean variable.
Interpretation: a particular assignment to arguments of a Boolean Expression.
Examples:

Logically equivalent

Two Boolean expressions $E_1, E_2 \in \mathcal{F}$ are *logically equivalent* if $v(E_1) = v(E_2)$ for all possible interpretations v. We write this as $E_1 \equiv E_2$.
(Informally, we say they are logically equivalent if their truth-table have the same output column.)

Some examples from Lecture 2:

- $a \oplus b$ is logically equivalent to $(a \vee b) \wedge \neg(a \wedge b)$.
- $(a \vee b) \wedge c$ is logically equivalent to $(a \wedge c) \vee (b \wedge c)$.
- $(a \wedge b) \vee c$ is logically equivalent to $(a \vee c) \wedge (b \vee c)$.

Another very important, though obvious one is:
$\neg(\neg a)$ is logically equivalent to a.

The most important identities in this module are the ones that show how AND, OR and NOT relate to each other. They are:

de Morgan's laws

(a) $\neg(a \wedge b) \equiv \neg a \vee \neg b$

(b) $\neg(a \vee b) \equiv \neg a \wedge \neg b$

We will prove (b) and leave (a) as an exercise.

In Lecture 1 we considered three 2-place operators ($\wedge$, $\vee$, $\oplus$). Of the remaining **thirteen**, several have special significance in Mathematics, Computing and Electronics. These are

1. *nor*, written as $a \downarrow b$;
2. *nand*, written as $a \uparrow b$;
3. *equivalence*, written as $a \leftrightarrow b$.
4. *implication*, written as $a \rightarrow b$, and read as "a implies b";

The logic table for "NOR" is:

It is important to note how it can be written in term of OR and NOT. So we say that

$$a \downarrow b \text{ is } \textit{logically equivalent} \text{ to } \neg(a \vee b).$$

The logic table for "NAND" is:

Again, this can be written in term of AND and NOT. So:

$a \uparrow b$ is *logically equivalent* to $\neg(a \wedge b)$.

Warning: the word "equivalence" is about to be used in a context that is different from "logical equivalence" given above.

The logic table for the "equivalence" operator, written as $\leftrightarrow$ is:

Can this be written in terms of other operators?

Our last example is probably the most important and confusing. It is called "implication" or "material implication", written as $a \rightarrow b$, and read as "a implies b". The table is:

IMPORTANT: there are two ways (that I can think of) to express this operator in "natural language", and which have can also be determined by checking logic tables:

They are:...

1. The *reverse equivalence* operator $\leftarrow$, has the table

a	b	$a \leftarrow b$
F	F	T
F	T	F
T	F	T
T	T	T

   (a) Show how it can be expressed using operators in the set $\{\neg, \rightarrow\}$.
   (b) Show how it can be expressed using $\{\neg, \vee\}$.
   (c) Show how it can be expressed using $\{\neg, \wedge\}$.

2. Show that $a \downarrow a \equiv \neg a$; Show that $a \uparrow a = \neg a$;
3. Of the following binary operators, which commute and which do not? (Recall, the 2-place operator $\star$ commutes if $a \star b \equiv b \star a$).
   (a) exclusive-or $\oplus$
   (b) NAND $\uparrow$
   (c) material implication $\Longrightarrow$
   (d) equivalence $\leftrightarrow$
4. Show that $a \rightarrow a \equiv T$ and that $a \leftrightarrow a \equiv T$.
5. Show that $a \rightarrow b \equiv \neg(a \wedge \neg b)$