CS304 – CS310

Mathematical and Logical Aspects of Computing

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

Tuesday, 11^{th} Sep 2012

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)
- *nor*, written as $a \downarrow b$;
- *nand*, written as $a \uparrow b$; (sometimes: $|$ and called the "Scheffer stroke").

. . . .

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

1. *equivalence*, written as $a \leftrightarrow b$.
2. *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:

- $(a \vee b) \wedge c$ is logically equivalent to $(a \wedge c) \vee (b \wedge c)$.
- de Morgan's laws
  (a) $\neg(a \wedge b) \equiv \neg a \vee \neg b$
  (b) $\neg(a \vee b) \equiv \neg a \wedge \neg b$

Equivalence $\leftrightarrow$

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:

This operator is sometimes known as "*if and only if*", and also is often denoted as $a \Leftrightarrow b$.

Can this be written in terms of other operators?

(Material) Implication $\longrightarrow$

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

Wason's selection task

See Chapter 4 of Chiswell and Hodges "*Mathematical Logic*"

You will be shown 4 cards. Each has a number written on one side and a letter on the other.

You are shown a statement S about the cards. Then you must answer the following question:

> *"Which card or cards must I turn over in order to check whether the statement S is true?"*

The cards are:

The statement is:

> *S: If a card has a vowel one on side, it has an even number on the other.*

Venn Diagrams

It has been mentioned that there is a close relationship between the *conjunction* in logic, and *intersection* in set theory. This can be demonstrated using a Venn diagram.

Example:

Similarly, there is a relationship between

- *disjunction* and *union*.
- *negation* and *compliment*

Example:

Exercises

1. Sketch the Venn Diagram for all operators mentioned in this lecture.
2. The *reverse equivalence* operator $\leftarrow$, has the table

F	F	T
F	T	F
T	F	T
T	T	T

   (a) Show how it can be expressed using $\leftarrow$.
   (b) Sketch the Venn Diagram for it.
3. Show that $a \downarrow a \equiv \neg a$; Show that $a \uparrow a = \neg a$;
4. Of the following binary operators, which commute ad 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 $\implies$
   (d) equivalence $\leftrightarrow$
5. Show that $a \rightarrow a \equiv T$ and that $a \leftrightarrow a \equiv T$.
6. Show that $a \rightarrow b \equiv \neg(a \wedge \neg b)$