Truth Tables Equivalence Validity

Truth Assignments

A truth assignment assigns a value T or F to each propositional variable. Computer scientists call assignment of values to variables an environment. If we know the environment, we can find the value of a propositional formula.

Evaluation in an Environment

Example: Suppose environment, v, assigns
v(P) = T, v(Q)= T, v(R) = F.
Truth value of
(NOT(P AND Q)) OR (R XOR NOT(Q))
F T T T F F F F T

Equivalence

Two propositional formulas are equivalent iff they have the same truth value in all environments.

DeMorgan's Law

$\overline{P \text{ OR } Q}$ equiv to $\overline{P}$ AND $\overline{Q}$

P	Q	NOT(P OR Q)
T	T	F
T	F	F
F	T	F
F	F	T

$\overline{P}$	AND	$\overline{Q}$
F	F	F
F	F	T
T	F	F
T	T	T

DeMorgan's Law

$\overline{P \text{ OR } Q}$ equiv to $\overline{P}$ AND $\overline{Q}$

P	Q	NOT(P OR Q)
T	T	F
T	F	F
F	T	F
F	F	T

$\overline{P}$	AND	$\overline{Q}$
F	F	F
F	F	T
T	F	F
T	T	T

Same final column, so equivalent
-- proof by Truth Table

Definition of IFF

The value of (P IFF Q) is T iff
P and Q have the *same* truth value.

Truth Table for IFF

P	Q	P IFF Q
T	T	T
T	F	F
F	T	F
F	F	T

Satisfiability & Validity

A formula is satisfiable iff it is true in some environment.

A formula is valid iff it is true in all environments.

Satisfiability & Validity

satisfiable: P, NOT(P)
not satisfiable: (P AND NOT(P))
valid: (P OR NOT(P))

Equivalence & Validity

G and H are equivalent exactly when (G IFF H) is valid

Verifying Valid, Satisfiable

Truth table size doubles with each additional variable --exponential growth. Makes truth tables impossible when there are hundreds of variables. (In current digital circuits, there are millions of variables.)

Efficient Test for Satisfiability?

The P = NP? question is equivalent to asking if there is an "efficient" (polynomial rather than exponential time) procedure to check satisfiability.

SAT versus VALID

To check that G is valid,
can check that
NOT(G) is not satisfiable.
So checking for one is
equally difficult (or easy)
as checking for the other.

MIT OpenCourseWare
http://ocw.mit.edu

6.042J / 18.062J Mathematics for Computer Science
Spring 2015

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.