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
\[ \text{NOT}(P \ \text{AND} \ Q) \ \text{OR} \ (R \ \text{XOR} \ \text{NOT}(Q)) \]
\[
\begin{array}{c|c|c|c|c|c|c|}
\text{} & T & T & T & F & F & F & T \\
\hline
P & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
Q & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
\end{array}
\]

Equivalence

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

\[ \overline{P \lor Q} \text{ equiv to } \overline{P} \land \overline{Q} \]

\[
\begin{array}{cccc}
P & Q & \overline{P} \lor Q & \overline{P} \land \overline{Q} \\
T & T & F & T \\
T & F & F & F \\
F & T & F & T \\
F & F & T & T \\
\end{array}
\]

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.

\[
\begin{array}{ccc}
P & Q & P \iff Q \\
T & T & T \\
T & F & F \\
F & T & F \\
F & F & T \\
\end{array}
\]

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: \( P, \ \text{NOT}(P) \)
not satisfiable: \( (P \ \text{AND} \ \text{NOT}(P)) \)
valid: \( (P \ \text{OR} \ \text{NOT}(P)) \)

Equivalence & Validity

G and H are equivalent exactly when \( (G \ \text{IFF} \ H) \) 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 \( P = \text{NP?} \) 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 $\text{NOT}(G)$ is not satisfiable. So checking for one is equally difficult (or easy) as checking for the other.