# Logic

## Boolean Satisfiability

Given a Boolean formula $\varphi(x_1,\ldots,x_n)$, the **Boolean satisfiability problem** asks whether there exists some choice of *true* and *false* values for $x_1 \ldots, x_n$ such that $\varphi(x_1,\ldots,x_n) = true$. This choice of variables is called a **satisfying assignment**, and the formula is said to be **satisfiable**. The satisfiability problem is known to be computationally difficult and was one of the first problems shown to be NP-complete.

Maple 2016 introduces new efficient heuristics for determining satisfiability and testing equivalence of Boolean expressions.

$with(Logic):$

$Satisfiable(\ x \textbf{ and } y \textbf{ xor not } z)$

$$true$$

$Satisfy(x \textbf{ and } y \textbf{ xor not } z)$

$$\{x = false, y = false, z = false\}$$

$Satisfiable(x \textbf{ and } y \textbf{ and not } x)$

$$false$$

$formula := Random([s, t, u, v, w, x, y, z], clauses = 20, literals = 3, form = CNF)$

$$(s \lor t \lor y) \land (s \lor u \lor w) \land (s \lor w \lor \neg x) \land (s \lor y \lor z) \land (t \lor w \lor y) \land (t \lor \neg u \lor \neg w) \land (t \lor \neg v \lor \neg w) \land (u \lor w \lor z) \land (u \lor y \lor \neg z) \land (v \lor y \lor \neg u) \land (w \lor \neg t \lor \neg z) \land (x \lor \neg t \lor \neg w) \land (x \lor \neg t \lor \neg z) \land (z \lor \neg t \lor \neg v) \land (\neg s \lor \neg t \lor \neg w) \land (\neg s \lor \neg u \lor \neg w) \land (\neg s \lor \neg v \lor \neg y) \land (\neg t \lor \neg u \lor \neg z) \land (\neg t \lor \neg x \lor \neg z) \land (\neg u \lor \neg w \lor \neg y)$$

$Satisfy(\ formula)$

$$\{s = false, t = false, u = true, v = false, w = false, x = false, y = true, z = false\}$$

## Truth Tables

The Logic:-TruthTable command now returns a DataFrame with the truth assignments for a given formula.

$TruthTable(x \textbf{ and } y \textbf{ xor not } z)$

| | x | y | z | value |
|---|---|---|---|---|
| 1 | *false* | *false* | *false* | *true* |
| 2 | *false* | *false* | *true* | *false* |
| 3 | *false* | *true* | *false* | *true* |
| 4 | *false* | *true* | *true* | *false* |
| 5 | *true* | *false* | *false* | *true* |
| 6 | *true* | *false* | *true* | *false* |
| 7 | *true* | *true* | *false* | *false* |
| 8 | *true* | *true* | *true* | *true* |