Arbitrary boolean function (expressed by input/output table) \(\Rightarrow\) Equivalent Boolean expression using only addition, multiplication, complement operations.

The set of operations: addition, multiplication, complement if functionally complete.

Can we remove one of the operations + still remain functionally complete.

De Morgan: \(\overline{xy} = \overline{x} + \overline{y}\)

\[\Rightarrow xy = \overline{xy} = \overline{\overline{x} + \overline{y}}\]

\[= \overline{xy} \quad \text{Computes the product without the multiplication op}!\]

Apply within more complex expressions to eliminate all multiplication operations.

\[\overline{(x + y)z} = \overline{x + y} + \overline{z}\]

\[(xy + z)w = \overline{\overline{x + y} + \overline{z}}w\]

\[= \overline{x + y + z + \overline{w}}\]
The set $\mathbb{Z}$ addition, complement $\mathbb{Z}$ is functionally complete.

Arbitrary boolean function (expressed by inputs/outputs table) $\rightarrow$ Boolean expression using $+$, $\cdot$, $-$ operations.

$\downarrow$ Remove multiplication

Boolean expression using only addition $+$ complement.

What about the set $\mathbb{Z}$ multiplication, complement $\mathbb{Z}$?

De Morgan again!

$\overline{x + y} = \overline{x \cdot y}$

$x + y = \overline{x + y}^{\perp} = \overline{\overline{x} \cdot \overline{y}}$

Can use this rule to eliminate any addition operation in a Boolean expression.

$(x + y)z \equiv (\overline{x \cdot y})z$

$xy + z \equiv \overline{xy}z$

$xy + z(x+y) = xy + \overline{z} (\overline{x \cdot y})$
\[
\frac{\ldots}{x y - \frac{1}{2} (x \bar{y})}
\]

The set \( \{ \text{multiplication, complement} \} \) is functionally complete.

Arbitrary boolean function (expressed by input/output table) \( \rightarrow \) Boolean expression using \(+, -, \cdot, \cdot\overline{\cdot}\) operations.

\( \rightarrow \) Remove addition.

Boolean expression using only multiplication and complement.

Is the set \( \{ \text{addition, multiplication} \} \) functionally complete?

No. Can't compute \( \bar{x} \).

Is there a single operation which is functionally complete all by itself?

Yes!

\[
\text{NAND} = x \cdot \bar{y} = \bar{x} \bar{y}
\]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Only need to show how to compute $x$ and $xy$.

$\begin{array}{c|c|c|c}
\top & 0 & 1 \\
\hline
\top & x & x \oplus \overline{x} \\
\hline
0 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}$

$\overline{xy} = x \oplus y \\
xy = \frac{\overline{xy}}{x \oplus y}$

$\begin{array}{c|c|c|c|c}
\top & x & y & x \oplus y & (x \oplus y) \oplus (\overline{x} \oplus y) \\
\hline
0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 \\
\end{array}$
\[ \overline{xy} = \overline{x} \cdot \overline{y} \]

\[ \overline{xy} + \overline{z} = \overline{xy} \cdot \overline{z} = \overline{xy} \cdot \overline{z} \]

\[ \overline{(x \cdot y)^2} = \overline{xy} \div 2 \]

\[ \overline{((\overline{x} \cdot \overline{y})^2) \div ((\overline{x} \cdot \overline{y})^2)} \]

\[ \overline{((\overline{x} \cdot \overline{y})^2) \div ((\overline{x} \cdot \overline{y})^2)} \]

\[ \overline{(x \cdot x \cdot \overline{y})} \]

\[ ((\overline{x} \cdot \overline{y})^2) \div ((\overline{x} \cdot \overline{y})^2) \]
**Boolean Satisfiability**

**SAT**

**Input:** Boolean expression with input variables.

**Output:** Yes/No:

Is there a setting of the input variables that cause the boolean expression to evaluate to 1?

\[
f(x, y, z) = \overline{z} \cdot (x + \overline{y}) \cdot (\overline{z} + \overline{x}) \cdot \overline{y} = 1
\]

**Not Satisfiable.**

\[
f(x, y, z) = \frac{1}{2} \cdot (x + \overline{y}) \cdot (\overline{z} + \overline{x}) \cdot (y + \overline{z}) \cdot \overline{y} = 1
\]

\[
f(x, y, z) = (\overline{z} + x + \overline{y}) \cdot (y + \overline{x} + z) \cdot (\overline{z} + x) \cdot (\overline{x} + y + \overline{z})
\]

One way to check is by truth table:

...
\[ f(x, y, z) = \frac{1}{(x+y)(y+z)(x+z)(x+y+z)} \]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>( f(x, y, z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

If a Boolean expression has 200 variables, this will require the function to be evaluated as many as \( 2^{200} \) times.

1 nanosecond per evaluation:
... over a billion years!

Why is SAT important?

The language of Boolean logic can be used to express many important & practical computational problems.

Efficient algorithm for SAT \( \implies \) Efficient algorithm for all these problems.
Scheduling Classes.

Set of classes.
Set of time slots.

Certain pairs of classes can not be assigned to the same time slot.

Is there a way to schedule the classes that does not violate any of the constraints?

I : input to scheduling problem.
\( n \) : \# of classes.
\( m \) : \# of time slots.

Pairs \((i,j)\) class \(j\) and \(i\) can not be scheduled during the same time.

\[ E \rightarrow \text{E is satisfiable if and only if there is a feasible schedule for I.} \]
For example: 5 classes \( \{A, B, C, D, E\} \).

2 time slots

\((A, B) \quad (C, D) \quad (D, E) \quad (B, E) \quad (A, D)\).

Boolean variables:

\(X_{A1}, X_{B1}, X_{C1}, X_{D1}, X_{E1}\)

\(X_{A2}, X_{B2}, X_{C2}, X_{D2}, X_{E2}\).

\(X_{C2} = 1\) if class C is scheduled during time slot 2.

\[ \overline{X_{A2}} \cdot \overline{X_{B2}} \]

\[(X_{D1} + X_{D2}) (\overline{X_{D1}} + \overline{X_{D2}}) \]

D must be scheduled in at least one of the two time slots.

D cannot be scheduled in both time slots.

Constraint \((D, E)\): D and E can not both be scheduled in the same time slot:

\[(\overline{X_{D1}} + \overline{X_{E1}}) (\overline{X_{D2}} + \overline{X_{E2}}) \]

D and E can not both be scheduled in time slot 1.

D and E can not both be scheduled in time slot 2.
Put together in one big product:

$$(X_{A_1} + X_{A_2})(X_{A_1} + X_{A_2})(X_{B_1} + X_{B_2})(X_{B_1} + X_{B_2})$$

$$(X_{C_1} + X_{C_2})(X_{C_1} + X_{C_2})(X_{D_1} + X_{D_2})(X_{D_1} + X_{D_2})$$

$$(X_{E_1} + X_{E_2})(X_{E_1} + X_{E_2})$$

times:

$$(X_{A_1} + X_{D_1})(X_{A_2} + X_{B_2})(X_{C_1} + X_{D_1})(X_{C_2} + X_{D_2})$$

$$(X_{D_1} + X_{E_1})(X_{D_2} + X_{E_2})(X_{B_1} + X_{E_1})(X_{B_2} + X_{E_2})$$

$$(X_{A_1} + X_{D_1})(X_{A_2} + X_{D_2})$$