3SAT

\((A \lor B \lor \neg C) \land (\neg B \lor C \lor \neg D) \land (\neg A \lor \neg B \lor D) \land \ldots\)

Yes if \(\exists\) a satisfying assignment

\[A = T \quad C = T\]
\[B = F \quad D = T\]
Polynomial time verifier for 3SAT

**Describe**
- certificate \( w \)
- verifier alg. \( V \)

**Prove**
- \( w \) has poly-length
- \( V \) takes poly-time
- \( \exists w \) such that
  \[ V(x, w) = \text{Yes} \]
  iff \( x \in L \)
Aside on Boolean logic

if and only if (iff)  if, then
\[ p \iff q \equiv (p \implies q) \land (\neg p \implies \neg q) \]
\[ \equiv (p \implies q) \land (q \implies p) \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p \implies q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>\neg p</th>
<th>\neg q</th>
<th>\neg q \implies \neg p</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
3SAT verifier

\( w = \) an assignment of truth values to variables

\( A = T, B = F, C = T, D = T \)

V

- verify that \( w \) is a valid assignment
- go through formula \( \phi \), plug in assignments from \( w \). Say Yes if all clauses eval to T.
$3\text{SAT} \leq_p \text{IS}$

- poly # of black box calls
- poly time translation steps
- correctness

\[(A \lor B \lor \neg C) \land (\neg B \lor C \lor \neg D) \land (\neg A \lor \neg B \lor D) \land \cdots \quad K = \# \text{ clauses}\]
Why do we care about $\leq_p$?

$A \leq_p B$ if $\exists$ poly-time alg. for $B$ then $\exists$ poly-time alg. for $A$
The set of all decision problems for which there exists a polynomial time algorithm.

The set of all decision problems for which a polynomial time verifier exists.

\[ P \subseteq NP \]

\[ P \neq NP \quad P = NP \]