binary expressions:

theorems:

antitheorems:
binary expressions: represent anything that comes in two kinds

division: 

theorems: represent one kind

antithereoms: represent the other kind
binary expressions: represent anything that comes in two kinds
represent statements about the world (natural or constructed, real or imaginary)

theorems: represent one kind
represent true statements

antitheorems: represent the other kind
represent false statements
binary expressions: represent anything that comes in two kinds
represent statements about the world (natural or constructed, real or imaginary)
represent digital circuits

theorems: represent one kind
represent true statements
represent circuits with high voltage output

antitheorems: represent the other kind
represent false statements
represent circuits with low voltage output
binary expressions: represent anything that comes in two kinds

represent statements about the world (natural or constructed, real or imaginary)

represent digital circuits

represent human behavior

theorems: represent one kind

represent true statements

represent circuits with high voltage output

represent innocent behavior

antitheorems: represent the other kind

represent false statements

represent circuits with low voltage output

represent guilty behavior
0 operands  \top  \bot
0 operands  \( \top \ \bot \)
1 operand  \( \neg x \)
<table>
<thead>
<tr>
<th>Number of Operands</th>
<th>Symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 operands</td>
<td>$\top$, $\bot$</td>
</tr>
<tr>
<td>1 operand</td>
<td>$\neg x$</td>
</tr>
<tr>
<td>2 operands</td>
<td>$x \land y$, $x \lor y$, $x \Rightarrow y$, $x \Leftarrow y$, $x = y$, $x \neq y$</td>
</tr>
</tbody>
</table>
0 operands \( \top \) \( \bot \)

1 operand \( \neg x \)

2 operands \( x \land y \) \( x \lor y \) \( x \Rightarrow y \) \( x \Leftarrow y \) \( x = y \) \( x \neq y \)
0 operands \( \top \) \( \bot \)

1 operand \( \neg x \)

2 operands \( x \land y \) \( x \lor y \) \( x \Rightarrow y \) \( x \Leftarrow y \) \( x = y \) \( x \neq y \)
0 operands \( \top \quad \bot \)

1 operand \( \neg x \)

2 operands \( x \land y \quad x \lor y \quad x \Rightarrow y \quad x \Leftrightarrow y \quad x = y \quad x \neq y \)
0 operands  \( \top \)  \( \bot \)
1 operand  \( \neg x \)
2 operands  \( x \land y \)  \( x \lor y \)  \( x \Rightarrow y \)  \( x \Leftarrow y \)  \( x = y \)  \( x \neq y \)
0 operands \quad \top \quad \bot
1 operand \quad \neg x
2 operands \quad x \land y \quad x \lor y \quad x \Rightarrow y \quad x \Leftarrow y \quad x = y \quad x \neq y
0 operands  \( \top \)  \( \bot \)
1 operand  \( \neg x \)
2 operands  \( x \land y \)  \( x \lor y \)  \( x \Rightarrow y \)  \( x \iff y \)  \( x = y \)  \( x \neq y \)
0 operands \( \top \), \( \bot \)

1 operand \( \neg x \)

2 operands \( x \land y \), \( x \lor y \), \( x \Rightarrow y \), \( x \Leftarrow y \), \( x = y \), \( x \neq y \)

3 operands \( \text{if } x \text{ then } y \text{ else } z \text{ fi} \)
0 operands \( \top \), \( \bot \)

1 operand \( \neg x \)

2 operands \( x \land y \), \( x \lor y \), \( x \Rightarrow y \), \( x \Leftrightarrow y \), \( x = y \), \( x \neq y \)

3 operands \textbf{if} \( x \) \textbf{then} \( y \) \textbf{else} \( z \) \textbf{fi}

precedence and parentheses
0 operands \( \top \quad \bot \)

1 operand \( \neg x \)

2 operands \( x \land y \quad x \lor y \quad x \Rightarrow y \quad x \Leftarrow y \quad x = y \quad x \neq y \)

3 operands \( \text{if } x \text{ then } y \text{ else } z \text{ fi} \)

precedence and parentheses

associative operators: \( \land \quad \lor \quad = \quad \pm \)

\[ x \land y \land z \text{ means either } (x \land y) \land z \text{ or } x \land (y \land z) \]

\[ x \lor y \lor z \text{ means either } (x \lor y) \lor z \text{ or } x \lor (y \lor z) \]
0 operands  \[ \top \quad \bot \]
1 operand  \[ \neg x \]
2 operands  \[ x \land y \quad x \lor y \quad x \Rightarrow y \quad x \Leftarrow y \quad x = y \quad x \neq y \]
3 operands  \[ \textbf{if} \ x \ \textbf{then} \ y \ \textbf{else} \ z \ \textbf{fi} \]

precedence and parentheses

associative operators:  \[ \land \quad \lor \quad = \quad \neq \]

\[ x \land y \land z \] means either \((x \land y) \land z\) or \(x \land (y \land z)\)

\[ x \lor y \lor z \] means either \((x \lor y) \lor z\) or \(x \lor (y \lor z)\)

continuing operators:  \[ \Rightarrow \quad \Leftarrow \quad = \quad \neq \]

\[ x = y = z \] means \(x = y \land y = z\)

\[ x \Rightarrow y \Rightarrow z \] means \((x \Rightarrow y) \land (y \Rightarrow z)\)
0 operands \[ \top \ \bot \]

1 operand \[ \neg x \]

2 operands \[ x \land y \ x \lor y \ x \Rightarrow y \ x \Leftarrow y \ x = y \ x \neq y \]

3 operands \[ \text{if } x \text{ then } y \text{ else } z \text{ fi} \]

precedence and parentheses

associative operators: \[ \land \ \lor = \ \pm \]

\[ x \land y \land z \text{ means either } (x \land y) \land z \text{ or } x \land (y \land z) \]

\[ x \lor y \lor z \text{ means either } (x \lor y) \lor z \text{ or } x \lor (y \lor z) \]

continuing operators: \[ \Rightarrow \ \Leftarrow = \ \pm \]

\[ x = y = z \text{ means } x = y \land y = z \]

\[ x \Rightarrow y \Rightarrow z \text{ means } (x \Rightarrow y) \land (y \Rightarrow z) \]

big operators: \[ = \Rightarrow \Leftarrow \]

same as \[ = \Rightarrow \Leftarrow \text{ but later precedence} \]

\[ x = y \Rightarrow z \text{ means } (x = y) \land (y \Rightarrow z) \]
truth tables

\[
\begin{array}{cc}
\top & \bot \\
\neg & \bot \quad \top
\end{array}
\]

\[
\begin{array}{cccc}
T & T & T & \bot \\
\wedge & T & \bot & \bot & \bot \\
\vee & T & T & T & \bot \\
\Rightarrow & T & \bot & T & T \\
\Leftarrow & T & T & \bot & T \\
\equiv & T & \bot & \bot & T \\
\neq & \bot & T & T & \bot \\
\end{array}
\]

if then else fi

\[
\begin{array}{cccccccccccc}
T & T & T & T & T & T & T & T & T & T & T & T \\
\end{array}
\]
### truth tables

<table>
<thead>
<tr>
<th></th>
<th>⊤</th>
<th>⊥</th>
</tr>
</thead>
<tbody>
<tr>
<td>¬</td>
<td>⊥</td>
<td>⊤</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>∧</th>
<th>⊤</th>
<th>⊥</th>
<th>⊥</th>
<th>⊥</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>∨</th>
<th>⊤</th>
<th>⊤</th>
<th>⊤</th>
<th>⊥</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>⊥</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>⇒</th>
<th>⊤</th>
<th>⊥</th>
<th>⊤</th>
<th>⊤</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>⊥</td>
<td>⊤</td>
<td>⊤</td>
<td>⊤</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>⇔</th>
<th>⊤</th>
<th>⊤</th>
<th>⊥</th>
<th>⊤</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>⊥</td>
<td>⊤</td>
<td>⊤</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>=</th>
<th>⊤</th>
<th>⊥</th>
<th>⊥</th>
<th>⊤</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>⊥</td>
<td>⊥</td>
<td>⊤</td>
<td>⊤</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>≠</th>
<th>⊤</th>
<th>T</th>
<th>T</th>
<th>⊥</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊥</td>
<td>T</td>
<td>T</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>

| if then else fi | T T T T T T T T T T T T T T T T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | T | ⊥ | ⊥ | ⊥ | T | ⊥ | ⊥ | T | ⊥ | ⊥ |

21/81
truth tables

<table>
<thead>
<tr>
<th>symbol</th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>¬</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>symbol</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>∧</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>∨</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>⇒</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>⇐</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>=</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>≠</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

if then else fi

<table>
<thead>
<tr>
<th>condition</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>if then else fi</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
# Truth Tables

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>¬</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>T T</th>
<th>T F</th>
<th>F T</th>
<th>F F</th>
</tr>
</thead>
<tbody>
<tr>
<td>∧</td>
<td>T T</td>
<td>T F</td>
<td>F F</td>
<td>F F</td>
</tr>
<tr>
<td>∨</td>
<td>T T</td>
<td>T T</td>
<td>T F</td>
<td>F F</td>
</tr>
<tr>
<td>⇒</td>
<td>T F</td>
<td>T T</td>
<td>F T</td>
<td>F F</td>
</tr>
<tr>
<td>⇔</td>
<td>T T</td>
<td>F F</td>
<td>T F</td>
<td>F F</td>
</tr>
<tr>
<td>=</td>
<td>T T</td>
<td>F F</td>
<td>T F</td>
<td>F F</td>
</tr>
<tr>
<td>≠</td>
<td>F F</td>
<td>T F</td>
<td>F F</td>
<td>T F</td>
</tr>
</tbody>
</table>

**if then else fi**

<table>
<thead>
<tr>
<th></th>
<th>T T T</th>
<th>T T F</th>
<th>T F T</th>
<th>T F F</th>
<th>F T T</th>
<th>F T F</th>
<th>F F T</th>
<th>F F F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>
### truth tables

<table>
<thead>
<tr>
<th>(\neg)</th>
<th>(\top)</th>
<th>(\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\bot)</td>
<td>(\top)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\land)</th>
<th>(\top)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\top)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\lor)</th>
<th>(\top)</th>
<th>(\top)</th>
<th>(\top)</th>
<th>(\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\top)</td>
<td>(\top)</td>
<td>(\top)</td>
<td>(\top)</td>
<td>(\top)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\Rightarrow)</th>
<th>(\top)</th>
<th>(\bot)</th>
<th>(\top)</th>
<th>(\top)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\top)</td>
<td>(\bot)</td>
<td>(\top)</td>
<td>(\top)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\Leftarrow)</th>
<th>(\top)</th>
<th>(\top)</th>
<th>(\bot)</th>
<th>(\top)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\top)</td>
<td>(\top)</td>
<td>(\bot)</td>
<td>(\top)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\Rightarrow)</th>
<th>(\top)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\bot)</td>
<td>(\top)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\oplus)</th>
<th>(\bot)</th>
<th>(\top)</th>
<th>(\top)</th>
<th>(\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\bot)</td>
<td>(\top)</td>
<td>(\top)</td>
<td>(\bot)</td>
<td></td>
</tr>
</tbody>
</table>

### if then else fi

<table>
<thead>
<tr>
<th>(\top\top\top)</th>
<th>(\top\top\bot)</th>
<th>(\top\bot\top)</th>
<th>(\top\bot\bot)</th>
<th>(\bot\top\top)</th>
<th>(\bot\top\bot)</th>
<th>(\bot\bot\top)</th>
<th>(\bot\bot\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\bot\top\top)</td>
<td>(\bot\top\bot)</td>
<td>(\bot\bot\top)</td>
<td>(\bot\bot\bot)</td>
<td>(\bot\bot\top)</td>
<td>(\bot\bot\bot)</td>
<td>(\bot\bot\top)</td>
<td>(\bot\bot\bot)</td>
</tr>
</tbody>
</table>
## Truth Tables

<table>
<thead>
<tr>
<th></th>
<th>(\top)</th>
<th>(\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\neg)</td>
<td>(\bot)</td>
<td>(\top)</td>
</tr>
</tbody>
</table>

| \(\land\)  | \(\top\) | \(\bot\) | \(\bot\) | \(\bot\) |
| \(\lor\)   | \(\top\) | \(\top\) | \(\top\) | \(\bot\) |
| \(\Rightarrow\) | \(\top\) | \(\bot\) | \(\top\) | \(\top\) |
| \(\Leftarrow\) | \(\top\) | \(\top\) | \(\bot\) | \(\top\) |
| \(\Leftarrow\) | \(\bot\) | \(\top\) | \(\top\) | \(\bot\) |

<table>
<thead>
<tr>
<th>(\text{if then else fi})</th>
<th>(\top)</th>
<th>(\top)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
<th>(\bot)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\top)</td>
<td>(\top)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>(\bot)</td>
</tr>
</tbody>
</table>
# truth tables

\[
\begin{array}{c|cc}
\neg & \top & \bot \\
\hline
\bot & \bot & \top \\
\end{array}
\]

\[
\begin{array}{c|ccccc}
\land & \top & \bot & \bot & \bot & \bot \\
\hline
\top & \top & \bot & \bot & \bot & \bot \\
\bot & \bot & \top & \top & \top & \top \\
\Rightarrow & \top & \bot & \top & \top & \top \\
\iff & \top & \top & \bot & \top \\
\equiv & \top & \bot & \bot & \top \\
\pm & \bot & \top & \top & \bot \\
\end{array}
\]

\[
\begin{array}{c|ccccc}
\text{if then else fi} & \top & \top & \bot & \bot & \top \\
\hline
\top & \top & \bot & \bot & \top & \bot & \bot & \bot & \bot & \bot & \bot \\
\end{array}
\]
### Truth Tables

#### Negation

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>¬</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

#### Conjunction

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
<th>T</th>
<th>F</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>∧</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

#### Disjunction

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>∨</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

#### Implication

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>⇒</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

#### Equivalence

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>⇔</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

#### XOR

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊕</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

#### If-Then-Else-End If

<table>
<thead>
<tr>
<th></th>
<th>T T T T</th>
<th>T T F</th>
<th>T F F</th>
<th>T F F</th>
<th>F F F</th>
<th>F F F</th>
<th>F T F</th>
<th>F T F</th>
<th>F T T</th>
<th>F T T</th>
<th>F T T</th>
<th>F T T</th>
<th>F T T</th>
</tr>
</thead>
<tbody>
<tr>
<td>if then else fi</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
# Truth Tables

## Negation ($\neg$)

<table>
<thead>
<tr>
<th>$\top$</th>
<th>$\bot$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bot$</td>
<td>$\top$</td>
</tr>
</tbody>
</table>

## Logical Connectives

<table>
<thead>
<tr>
<th>Operator</th>
<th>$\top$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\top$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\bot$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\land$</td>
<td>$\top$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\lor$</td>
<td>$\top$</td>
<td>$\top$</td>
<td>$\top$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\Rightarrow$</td>
<td>$\top$</td>
<td>$\bot$</td>
<td>$\top$</td>
<td>$\top$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\Leftarrow$</td>
<td>$\top$</td>
<td>$\top$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$=$</td>
<td>$\top$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$\neq$</td>
<td>$\bot$</td>
<td>$\top$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
</tbody>
</table>

## Conditional ($\text{if then else fi}$)

<table>
<thead>
<tr>
<th>$\top$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\bot$</th>
<th>$\bot$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\top$</td>
<td>$\top$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
</tr>
</tbody>
</table>
variables are for substitution (instantiation)
variables are for substitution (instantiation)

• add parentheses to maintain precedence
  
in $x \land y$ replace $x$ by $\bot$ and $y$ by $\bot \lor \top$    result: $\bot \land (\bot \lor \top)$
variables are for substitution (instantiation)

• add parentheses to maintain precedence
  \[ x \land y \] replace \( x \) by \( \bot \) and \( y \) by \( \bot \lor \top \)
  result: \( \bot \land (\bot \lor \top) \)

• every occurrence of a variable must be replaced by the same expression
  \[ x \land x \] replace \( x \) by \( \bot \)
  result: \( \bot \land \bot \)
variables are for substitution (instantiation)

• add parentheses to maintain precedence

\[ \text{in } x \land y \text{ replace } x \text{ by } \bot \text{ and } y \text{ by } \bot \lor \top \quad \text{result: } \bot \land (\bot \lor \top) \]

• every occurrence of a variable must be replaced by the same expression

\[ \text{in } x \land x \text{ replace } x \text{ by } \bot \quad \text{result: } \bot \land \bot \]

• different variables can be replaced by the same expression or different expressions

\[ \text{in } x \land y \text{ replace } x \text{ by } \bot \text{ and } y \text{ by } \bot \quad \text{result: } \bot \land \bot \]
variables are for substitution (instantiation)

• add parentheses to maintain precedence
  
  \[x \land y\] replace \(x\) by \(\bot\) and \(y\) by \(\bot \lor \top\)   result: \(\bot \land (\bot \lor \top)\)

• every occurrence of a variable must be replaced by the same expression

  \[x \land x\] replace \(x\) by \(\bot\)   result: \(\bot \land \bot\)

• different variables can be replaced by the same expression or different expressions

  \[x \land y\] replace \(x\) by \(\bot\) and \(y\) by \(\bot\)   result: \(\bot \land \bot\)

  \[x \land y\] replace \(x\) by \(\top\) and \(y\) by \(\bot\)   result: \(\top \land \bot\)
new binary expressions

(the grass is green)

(the sky is green)

(there is life elsewhere in the universe)

(intelligent messages are coming from space)
new binary expressions

(the grass is green)

(the sky is green)

(there is life elsewhere in the universe)

(intelligent messages are coming from space)

$1 + 1 = 2$

$0 / 0 = 5$
new binary expressions

(the grass is green)

(the sky is green)

(there is life elsewhere in the universe)

(intelligent messages are coming from space)

\[ 1 + 1 = 2 \]

\[ 0 / 0 = 5 \]

\[ \frac{72}{543} \]

\[ \frac{742}{36} \]

\[ \frac{757}{81} \]

**consistent**: no binary expression is both a theorem and an antitheorem

(no overclassified expressions)
new binary expressions

(the grass is green)
(the sky is green)
(there is life elsewhere in the universe)
(intelligent messages are coming from space)

1 + 1 = 2
0 / 0 = 5

consistent: no binary expression is both a theorem and an antitheorem
(no overclassified expressions)

complete: every fully instantiated binary expression is either a theorem or an antitheorem
(no unclassified expressions)
Proof Rules

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.
Proof Rules

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

\[ x + y = y + x \]

is a mathematical expression
Proof Rules

**Axiom Rule** If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

\[ x + y = y + x \]

is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend

on the order in which you put them together
Proof Rules

**Axiom Rule** If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antithorem.

\[ x + y = y + x \]

is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend

on the order in which you put them together

is an axiom
**Proof Rules**

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

\[x + y = y + x\]  

is a mathematical expression  

represents a truth in an application such that  

when you put quantities together, the total quantity does not depend  

on the order in which you put them together

is an axiom

is a theorem
Proof Rules

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antithorem.

\[ x + y = y + x \]

is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend

on the order in which you put them together

is an axiom

is a theorem

is equivalent to    \( \top \)
Proof Rules

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

\[ x + y = y + x \]

is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend on the order in which you put them together

is an axiom

is a theorem

is equivalent to \( \top \)

\[ x + y = y + x \]

is true  \( \text{(not really)} \)
Proof Rules

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

    If a binary expression is an antiaxiom, then it is an antitheorem.
**Proof Rules**

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

axiom:  \( \top \)

antiaxiom:  \( \bot \)
Proof Rules

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

axiom: \( \top \)

antiaxiom: \( \bot \)

axiom: (the grass is green)

antiaxiom: (the sky is green)
**Proof Rules**

**Axiom Rule**  If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

axiom: \( \top \)

antiaxiom: \( \bot \)

axiom: (the grass is green)

antiaxiom: (the sky is green)

axiom: (intelligent messages are coming from space)

\( \Rightarrow \) (there is life elsewhere in the universe)
Proof Rules

Axiom Rule  If a binary expression is an axiom, then it is a theorem.
If a binary expression is an antiaxiom, then it is an antitheorem.

axiom: \( \top \)
antiaxiom: \( \bot \)
axiom: (the grass is green)
antiaxiom: (the sky is green)
axiom: (intelligent messages are coming from space)
\[\Rightarrow\] (there is life elsewhere in the universe)

Evaluation Rule  If all the binary subexpressions of a binary expression are classified,
then it is classified according to the truth tables.
Completion Rule  If a binary expression contains unclassified binary subexpressions,

and all ways of classifying them place it in the same class, then it is in that class.
Proof Rules

**Completion Rule** If a binary expression contains unclassified binary subexpressions, and all ways of classifying them place it in the same class, then it is in that class.

theorem: (there is life elsewhere in the universe) ∨ T
Proof Rules

**Completion Rule** If a binary expression contains unclassified binary subexpressions, and all ways of classifying them place it in the same class, then it is in that class.

theorem: \( \text{(there is life elsewhere in the universe)} \lor \top \)

theorem: \( \text{(there is life elsewhere in the universe)} \lor \lnot(\text{there is life elsewhere in the universe}) \lor \top \)
Proof Rules

**Completion Rule** If a binary expression contains unclassified binary subexpressions, and all ways of classifying them place it in the same class, then it is in that class.

\[
\text{theorem: } \ (\text{there is life elsewhere in the universe}) \lor \top
\]

\[
\text{theorem: } \ (\text{there is life elsewhere in the universe}) \\
\lor \ (\neg \text{(there is life elsewhere in the universe)}
\]

\[
\text{antithorem: } \ (\text{there is life elsewhere in the universe}) \\
\land \ (\neg \text{(there is life elsewhere in the universe)}
\]
Proof Rules

Consistency Rule  If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.
Proof Rules

**Consistency Rule** If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that $x$ and $x \Rightarrow y$ are theorems. What is $y$?
Proof Rules

**Consistency Rule**  If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that \( x \) and \( x \Rightarrow y \) are theorems. What is \( y \)?

If \( y \) were an antitheorem, then by the Evaluation Rule, \( x \Rightarrow y \) would be an antitheorem. That would be inconsistent. So \( y \) is a theorem.
**Consistency Rule**  If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that $x$ and $x \Rightarrow y$ are theorems. What is $y$?

If $y$ were an antitheorem, then by the Evaluation Rule, $x \Rightarrow y$ would be an antitheorem. That would be inconsistent. So $y$ is a theorem.

We are given that $\neg x$ is a theorem. What is $x$?
**Proof Rules**

**Consistency Rule** If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that $x$ and $x \Rightarrow y$ are theorems. What is $y$?

If $y$ were an antitheorem, then by the Evaluation Rule, $x \Rightarrow y$ would be an antitheorem. That would be inconsistent. So $y$ is a theorem.

We are given that $\neg x$ is a theorem. What is $x$?

If $x$ were a theorem, then by the Evaluation Rule, $\neg x$ would be an antitheorem. That would be inconsistent. So $x$ is an antitheorem.
Proof Rules

**Consistency Rule** If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that $x$ and $x \Rightarrow y$ are theorems. What is $y$?

If $y$ were an antitheorem, then by the Evaluation Rule, $x \Rightarrow y$ would be an antitheorem.

That would be inconsistent. So $y$ is a theorem.

We are given that $\neg x$ is a theorem. What is $x$?

If $x$ were a theorem, then by the Evaluation Rule, $\neg x$ would be an antitheorem.

That would be inconsistent. So $x$ is an antitheorem.

No need to talk about antiaxioms and antitheorems.
Proof Rules

**Instance Rule**  If a binary expression is classified,

then all its instances have that same classification.
Proof Rules

**Instance Rule** If a binary expression is classified,

then all its instances have that same classification.

axiom: \( x = x \)
Proof Rules

**Instance Rule**  If a binary expression is classified,

then all its instances have that same classification.

axiom: \[ x = x \]

theorem: \[ x = x \]
Proof Rules

**Instance Rule**  If a binary expression is classified,

then all its instances have that same classification.

axiom: \[ x = x \]

theorem: \[ x = x \]

theorem: \[ \top = \bot \lor \bot \equiv \top = \bot \lor \bot \]
**Proof Rules**

**Instance Rule**  If a binary expression is classified,

then all its instances have that same classification.

axiom: \( x = x \)

theorem: \( x = x \)

theorem: \( \top = \bot \lor \bot = \top \lor \bot \)

theorem: (intelligent messages are coming from space)

= (intelligent messages are coming from space)
**Proof Rules**

**Instance Rule**  If a binary expression is classified,

then all its instances have that same classification.

axiom:  \( x = x \)

theorem:  \( x = x \)

theorem:  \( \top = \bot \lor \bot \Rightarrow \top = \bot \lor \bot \)

theorem:  (intelligent messages are coming from space)

\( = \) (intelligent messages are coming from space)

Classical Logic:  all five rules

Constructive Logic:  not Completion Rule

Evaluation Logic:  neither Consistency Rule nor Completion Rule
Expression and Proof Format

\( a \land b \lor c \)
Expression and Proof Format

\( a \land b \lor c \quad \text{NOT} \quad a \land b \lor c \)
Expression and Proof Format

\( a \land b \lor c \quad \text{NOT} \quad a \land b \lor c \)

\( (\text{first part} \quad \land \quad \text{second part} \quad ) \)
Expression and Proof Format

\[ a \land b \lor c \quad \text{NOT} \quad a \land b \lor c \]

\[
( \quad \text{first part} \\
\land \quad \text{second part} \quad )
\]

C and Java convention

```java
while (something) {
    various lines
    in the body
    of the loop
}
```
Expression and Proof Format

\( a \land b \lor c \quad \textbf{NOT} \quad a \land b \lor c \)

( \textit{first part} \\
\land \quad \textit{second part} )
Expression and Proof Format

\[ a \land b \lor c \] \hspace{1cm} \text{NOT} \hspace{1cm} \neg a \land b \lor c

( \text{first part} \land \text{second part} )

\text{first part} = \text{second part}
Expression and Proof Format

\[ a \land b \lor c \quad \text{NOT} \quad a \land b \lor c \]

( \text{first part} \land \text{second part} )

\text{first part} = \text{second part}

expression0

= expression1

= expression2

= expression3
Expression and Proof Format

\[ a \land b \lor c \quad \text{NOT} \quad a \land b \lor c \]

\[
( \text{first part} \\
\land \text{second part} )
\]

\[
\text{first part} = \text{second part}
\]

\[
\text{expression0} = \text{expression1} \\
\text{means} \quad \land \text{expression1} = \text{expression2} \\
\text{means} \quad \land \text{expression2} = \text{expression3}
\]
Expression and Proof Format

\[ a \land b \lor c \quad \text{NOT} \quad a \land b \lor c \]

( first part
\land \quad \text{second part} \quad )

first part
= second part

expression0
= expression1
= expression2
= expression3
Expression and Proof Format

\( \text{first part} \land \text{second part} \)

first part

\[ \text{first part} = \text{second part} \]

= expression0

= expression1

= expression2

= expression3
Expression and Proof Format

Prove $a \land b \Rightarrow c = a \Rightarrow (b \Rightarrow c)$
Expression and Proof Format

Prove \( a \land b \Rightarrow c = a \Rightarrow (b \Rightarrow c) \)

\[
\begin{align*}
   a \land b & \Rightarrow c \\
= & \quad \neg(a \land b) \lor c \\
= & \quad \neg a \lor \neg b \lor c \\
= & \quad a \Rightarrow \neg b \lor c \\
= & \quad a \Rightarrow (b \Rightarrow c)
\end{align*}
\]

Material Implication

Duality

Material Implication

Material Implication
Expression and Proof Format

Prove \( a \land b \implies c = a \implies (b \implies c) \)

\[
\begin{align*}
a \land b & \implies c \\
= & \quad \neg (a \land b) \lor c \quad \text{Material Implication} \\
= & \quad \neg a \lor \neg b \lor c \quad \text{Duality} \\
= & \quad a \implies \neg b \lor c \quad \text{Material Implication} \\
= & \quad a \implies (b \implies c) \quad \text{Material Implication}
\end{align*}
\]

Material Implication: \( a \implies b = \neg a \lor b \)
Prove $a \land b \Rightarrow c = a \Rightarrow (b \Rightarrow c)$

\[
\begin{align*}
\quad & a \land b \Rightarrow c \\
\quad \; & = \neg (a \land b) \lor c \\
\quad \; & = \neg a \lor \neg b \lor c \\
\quad \; & = a \Rightarrow \neg b \lor c \\
\quad \; & = a \Rightarrow (b \Rightarrow c)
\end{align*}
\]

**Material Implication:**

\[a \Rightarrow b = \neg a \lor b\]

**Instance of Material Implication:**

\[a \land b \Rightarrow c = \neg (a \land b) \lor c\]
Expression and Proof Format

Prove \( a \land b \implies c = a \implies (b \implies c) \)

\[
\begin{align*}
a \land b \implies c &= \neg(a \land b) \lor c \\
=\quad & \neg a \lor \neg b \lor c \\
=\quad & a \implies \neg b \lor c \\
=\quad & a \implies (b \implies c)
\end{align*}
\]
Expression and Proof Format

Prove \( a \land b \Rightarrow c = a \Rightarrow (b \Rightarrow c) \)

\[
\begin{align*}
& a \land b \Rightarrow c & \text{Material Implication} \\
= & \neg(a \land b) \lor c & \text{Duality} \\
= & \neg a \lor \neg b \lor c & \text{Material Implication} \\
= & a \Rightarrow \neg b \lor c & \text{Material Implication} \\
= & a \Rightarrow (b \Rightarrow c) & \\
\end{align*}
\]

\[
(a \land b \Rightarrow c = a \Rightarrow (b \Rightarrow c)) \quad \text{Material Implication 3 times}
\]

\[
\begin{align*}
& = (\neg(a \land b) \lor c = \neg a \lor (\neg b \lor c)) & \text{Duality} \\
& = (\neg a \lor \neg b \lor c = \neg a \lor \neg b \lor c) & \text{Reflexivity of } = \\
& = \top & \\
\end{align*}
\]