Propositional Operators

Propositional (Boolean) Logic

A proposition is either True or False

Example:

There are 6 regular solids.

False

Non-examples:

Wake up!
Where am I?
It's 3PM.

English to Math

Greeks carry Swords or Javelins

\[ G \rightarrow (S \lor J) \]

True even if a Greek carries both a Sword and a Javelin

English to Math

Greeks carry Bronze or Copper swords

\[ G \rightarrow (B \oplus C) \]

Bronze or Copper but not both
**Definition of OR**
The value of \((P \text{ OR } Q)\) is \(T\) iff
- \(P\) is \(T\), or \(Q\) is \(T\), or both are \(T\).

**Truth Table for OR**

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P OR Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

**Definition of XOR**
The value of \((P \text{ XOR } Q)\) is \(T\) iff
- exactly one of \(P\) and \(Q\) is \(T\).

**Truth Table for XOR**

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P XOR Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

**Definition of AND**
The value of \((P \text{ AND } Q)\) is \(T\) iff
- both \(P\) and \(Q\) are \(T\).

**Truth Table for AND**

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P AND 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>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

**Definition of NOT**
The value of \(\neg(P)\) is \(T\) iff
- the value of \(P\) is \(F\).

**Truth Table for NOT(P)**

<table>
<thead>
<tr>
<th>P</th>
<th>NOT(P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Other Applications

Java Logical Expressions:

```
OR        AND
if ((x>0) || (x <= 0 && y>100)) :
    (more code)
```

Digital Logic

1 ::= T
0 ::= F
• ::= AND
+ ::= OR
\(\overline{x}\) ::= NOT(x)

Application: Digital Logic

\(s ::= A \text{ XOR } B\)
\(c ::= A \text{ AND } B\)

half adder

Digital Logic

\(d ::= c_{\text{in}} \text{ XOR } s\)
\(c_{\text{out}} ::= (c_{\text{in}} \text{ AND } s) \text{ OR } c\)

full adder