

1

| | | |
|---------------------|---------------------|-------------------------------------|
| \vee \vee | \cdot \cdot | \triangleleft \triangleleft |
| \wedge \wedge | \diamond \diamond | \triangleright \triangleright |
| \amalg \amalg | \bullet \bullet | \bigtriangledown \bigtriangledown |
| \cap \cap | \circ \circ | \bigtriangleup \bigtriangleup |
| \cup \cup | \bigcirc \bigcirc | $*$ \ast |
| \uplus \uplus | \odot \odot | \star \star |
| \sqcap \sqcap | \ominus \ominus | \times \times |
| \sqcup \sqcup | \oplus \oplus | \div \div |
| \dagger \dagger | \oslash \oslash | \backslash \setminus |
| \ddagger \ddagger | \otimes \otimes | \wr \wr |
| \land \land | \pm \pm | |
| \lor \lor | \mp \mp | |

These commands produce the symbols for various binary operations. Binary operations are one of T_EX's classes of math symbols. T_EX puts different amounts of space around different classes of math symbols. When T_EX needs to break a line of text within a math formula, it will consider placing the break after a binary operation—but only if the operation is at the outermost level of the formula, i.e., not enclosed in a group.

In addition to these commands, T_EX also treats ‘+’ and ‘-’ as binary operations. It considers ‘/’ to be an ordinary symbol, despite the fact that mathematically it is a binary operation, because it looks better with less space around it.

Example:

```


$$z = x \div y \quad \text{if and only if} \quad z \times y = x \text{ and } y \neq 0$$


```

produces:

$$z = x \div y \quad \text{if and only if} \quad z \times y = x \text{ and } y \neq 0$$