Notation: Big operators

Very often, a formula requires you to combine a fixed collection of elements in a specific manner. For example, we may have a set of natural numbers $S := \{10, 35, 100, 275, 1000, 8883\}$ and want to combine them with some specific operation, say, $\oplus$. It does not matter what this operation does; $\oplus$ has no standardized meaning, it could be literally anything. Of course we could write out a formula like $10 \oplus 35 \oplus 100 \oplus 275 \oplus 1000 \oplus 8883$. But this has two downsides:

1. It is tedious for small sets and infeasible for large ones.
2. It presumes that we know the members of the set.

A more elegant solution is to use an indexed operator:

$$\bigoplus_{n \in S} n$$

An indexed operator has a subscripted condition that contains a variable (here n) and a specification of what value n can assume. In the example above, all instantiations of n must be members of S. One then instantiates all possible values of n and combines them with the operation defined by the operator.

This may still sound awfully abstract, but a few concrete examples will clarify things.

$\sum$: sum/addition

The most common operator is the sum operator $\sum$, which indicates addition (get it? **s**igma, **s**um). The elements to be summed are commonly drawn from a set or an interval.

Example 1

$$\sum_{n \in \{2,5,8,10\}} n = 2 + 5 + 8 + 10 = 25$$

Example 2

$$\sum_{1 \leq n \leq 10} n = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$

Sometimes, $\sum$ is used with a format where the subscript indicates the lowest of a range of values and a superscript specifies the cutoff point.

Example 3

$$\sum_{n=-5}^{7} = -5 + -4 + -3 + -2 + -1 + 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 = 13$$

The variable need not always occur by itself, it can be part of a larger expression.

Example 4

$$\sum_{n=1}^{3} 2 \times (n + 3) = 2 \times (1 + 3) + 2 \times (2 + 3) + 2 \times (3 + 3) = 8 + 10 + 18 = 36$$

Take care not to confuse the operator $\sum$ with the common alphabet symbol Σ. Sometimes both can show up in one and the same formula.

Example 5 Here is an example of a definition that mixes the sum operator $\sum$ with the alphabet symbol Σ:
As our alphabet Σ, we pick a random finite set of natural numbers. We define the ID-number of Σ as

$$\sum_{\sigma \in \Sigma} \sigma$$

$\prod$: product/multiplication

The operator $\prod$ works exactly like $\sum$, except that it denotes multiplication (so the letter **p**i is used for the **p**roduct operation).

Example 6
$$\prod_{n \in \{2,5,8,10\}} n = 2 \times 5 \times 8 \times 10 = 800$$

Example 7
$$\prod_{1 \leq n \leq 10} n = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10 = 3,628,800$$

Example 8
$$\prod_{n=-5}^{7} = -5 \times -4 \times -3 \times -2 \times -1 \times 0 \times 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 = 0$$

Example 9
$$\prod_{n=1}^{3} 2-(n \times 3) = (2-(1 \times 3)) \times (2-(2 \times 3)) \times (2-(3 \times 3)) = -1 \times -4 \times -7 = -28$$

Other examples

Many binary operators have big counterparts. This includes $\cap$ and $\cup$ in set theory, $\wedge$ and $\vee$ in logic, and non-descript operations like $\oplus$ and $\otimes$ in abstract algebra. Sometimes these operators can be nested or appear in sequence.

Example 10 Let $\mathscr{S}$ be a finite set of sets of natural numbers. More precisely, $\mathscr{S} := \{\{0,3,6\},\{2,3,9\},\{2,9\}\}$. Then

$$\prod_{S \in \mathscr{S}} \sum_{n \in S} n = \sum_{n \in \{0,3,6\}} n \times \sum_{n \in \{2,3,9\}} n \times \sum_{n \in \{2,9\}} n = (0+3+6) \times (2+3+9) \times (2+9) = 1386$$

Two important requirements

There are two important restrictions on the use of finite operators. First, they are only guaranteed to give a well-defined result if there are only finitely many substitutions for all variables.

Example 11 Without further stipulations, the following formula has no well-defined result:

$$\sum_{n>0} n = 1+2+3+4+5+... = ???$$

Second, the operation must be associative and commutative. Associativity means that the order of evaluation does not matter, whereas commutativity ensures that the operation does not care about the order of arguments. Together, these two properties guarantee that the result does not change depending on the order in which variables are replaced by values.

Example 12 Subtraction is neither associative nor commutative: $(5-2)-3 \neq 5-(2-3)$, and $5-2 \neq 2-5$. So there is no big operator version of subtraction because the results would depend on how variables are instantiated. For instance, $-_{n\in\{5,2\}} n$ could yield $5-2$ or $2-5$.