

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; the symbol \oplus has no standardized meaning in mathematics, it could be addition, multiplication, string concatenation, or a completely new operation that you just defined yourself — \oplus could be literally anything. The only thing that matters here is that we want to write an expression to show that we are combining all elements of S via \oplus .

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.

Σ : sum/addition

The most common operator is the sum operator Σ , which indicates addition (get it? sigma, sum). The elements to be summed are commonly drawn from a set or an interval.

Example

1

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

Note that it does not matter in which order we substitute values for n .

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

Example

2

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

Exercise

1

Write each one of the following expressions as a formula with the addition operator $+$.

$$5 - \sum_{1 \leq i \leq 3} i$$

$$\sum_{i=1} i + i + i$$

Exercise 2 Express each formula below in a more succinct format using the sum operator \sum .

- $20 + 21 + 22 + 23 + 24 + 25 + 26 + 27 + 28 + 29$
- $2 + 4 + 6 + 8 + 10$
- $30 + 15 + 3 + 10 + 2 + 20$
- $-10 - 9 - 8 - 7$

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

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

Exercise 3 Rewrite the formula below so that it uses the subscript/superscript format.

$$5 - \sum_{1 \leq i \leq 3} i$$

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

Example

4

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

Of course we could have also written this as follows:

$$\sum_{n \in \{8, 10, 18\}}$$

But this formula is just an idiosyncratic summation of three seemingly random numbers, whereas $\sum_{n=1}^3 2 \times (n + 3)$ captures a specific kind of systematicity. And this expression readily generalizes to values for n that are strictly greater than 3, which is not the case for the completely idiosyncratic $\sum_{n \in \{8, 10, 18\}}$.

Exercise 4 Consider once more the formulas below.

- $20 + 21 + 22 + 23 + 24 + 25 + 26 + 27 + 28 + 29$
- $2 + 4 + 6 + 8 + 10$
- $30 + 15 + 3 + 10 + 2 + 20$
- $-10 - 9 - 8 - 7$

Combine the sum operator with other operations in order to describe each formula in a more systematic way. Keep in mind that there are infinitely many ways this can be done, just pick the one that seems most plausible to you.

Quite generally, \sum can occur wherever addition may occur, so we can build some very complex formulas with this operator. We can even have multiple instances of \sum in the same expression.

Example

5

$$\begin{aligned} & \log_2 \left(\left(\sum_{0 \leq i \leq 5} 2^i \right) - 2 \times \sum_{j \in \{3,5,7,9\}} \frac{j}{2} \right) \\ &= \log_2 \left((2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5) - 2 \times \left(\frac{3}{2} + \frac{5}{2} + \frac{7}{2} + \frac{9}{2} \right) \right) \\ &= \log_2 (63 - 2 \times 12) \\ &= \log_2 (63 - 24) \\ &= \log_2 39 \\ &\approx 5.285 \end{aligned}$$

In fact, when we have multiple operators of this kind, variables of one can be dependent on the variables of another.

Example

6

$$\begin{aligned} & \sum_{i=1}^3 \sum_{j=0}^i i^j \\ &= \sum_{j=0}^1 1^j + \sum_{j=0}^2 2^j + \sum_{j=0}^3 3^j \\ &= 1^0 + 2^0 + 2^1 + 3^0 + 3^1 + 3^2 \\ &= 17 \end{aligned}$$

When \sum occurs as part of a larger expression, correct bracketing becomes very important. The scope of the sum operator extends rightward to its variable, but no further than that. Hence we have to use brackets to indicate parts that should be part of each summing step.

Example The two expressions below only differ in their bracketing and yield different results.

7

$$\sum_{n=1}^4 n + 5 = 1 + 2 + 3 + 4 + 5 = 15$$

$$\sum_{n=1}^4 (n + 5) = (1 + 5) + (2 + 5) + (3 + 5) + (4 + 5) = 35$$

The first expression could also be written as follows:

$$5 + \sum n = 1^4$$

Finally, 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 8 Here is an example of a definition that mixes the sum operator \sum with the alphabet symbol Σ :

Suppose that for some reason we need to define a new notion *ID-number* for alphabets that only contain natural numbers. Given such an alphabet Σ , its ID-number is

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

So if $\Sigma := \{2, 7, 201\}$, then its ID-number is

$$\sum_{\sigma \in \Sigma} \sigma = 2 + 7 + 201 = 210.$$

\prod : product/multiplication

The operator \prod works exactly like \sum , except that it denotes multiplication (so the letter pi is used for the product operation).

Example

9

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

Example

10

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

Example

11

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

Example

12

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

Just like \sum , \prod can occur as part of a larger formula with other operators.

Example

13

$$\begin{aligned} & \prod_{i=1}^3 \prod_{j=0}^i i^j \\ &= \prod_{j=0}^1 1^j \times \prod_{j=0}^2 2^j \times \prod_{j=0}^3 3^j \\ &= 1^0 \times 2^0 \times 2^1 \times 3^0 \times 3^1 \times 3^2 \\ &= 54 \end{aligned}$$

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.

Example

14

$$\bigcup_{S \in \{\{0,1,\dots,n\} \mid 3 \leq n \leq 5\}} S = \{0, 1, 2, 3\} \cup \{0, 1, 2, 3, 4\} \cup \{0, 1, 2, 3, 4, 5\} = \{0, 1, 2, 3, 4, 5\}$$

Example

15

$$\bigwedge_{n \in \{2,3,4\}} \neg \text{has-}n\text{-daughters}(x) = \\ \neg \text{has-2-daughters}(x) \wedge \neg \text{has-3-daughters}(x) \wedge \neg \text{has-4-daughters}(x)$$

And as before, all these operators can be nested or appear in sequence.

Example

16

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

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

Exercise

5

Suppose that $\mathcal{S} := \{\{0, 3, 6\}, \{2, 3, 9\}, \{2, 9\}\}$ as before. What is the value of the formula below, assuming that $i \oplus j = ij$ if $i \geq j$ and ji otherwise (for example, $8 \oplus 7 = 87$ and $3 \oplus 15 = 153$).

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

Exercise

6

Two important requirements

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

Example

17

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

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

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

18

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 $2 - 5$, which have very different results.

Even if we agreed on a convention for this, e.g. always substituting larger values before smaller ones, the resulting formula would still be ambiguous because subtraction is not associative. Contrast the following two:

$$-_{n \in \{5,3,2\}} n = (5 - 3) - 2 = 0$$

$$-_{n \in \{5,3,2\}} n = 5 - (3 - 2) = 4$$