

377 Show that we can define nat by fixed-point construction together with

(a) $\forall n: nat. 0 \leq n < n+1$

(b) $\exists m: nat. \forall n: nat. m \leq n < n+1$

After trying the question, scroll down to the solution.

(a) $\forall n: \text{nat} \cdot 0 \leq n < n+1$

§ We are given

$\text{nat} = 0, \text{nat}+1$

$\forall n: \text{nat} \cdot 0 \leq n < n+1$

and we must prove

$B = 0, B+1 \Rightarrow \text{nat}: B$

My first move is portation: given

$\text{nat} = 0, \text{nat}+1$

$\forall n: \text{nat} \cdot 0 \leq n < n+1$

$B = 0, B+1$

prove

$\text{nat}: B$

Now I am stuck; I have no idea how to proceed. So instead of a formal proof, I offer an informal proof. Consider bunch nat to be unknown. From nat fixed-point construction we can prove that $0, 1, 2$, and so on, are in nat . But maybe more elements are in nat . Maybe nat is all the integers

$\dots, -3, -2, -1, 0, 1, 2, 3, \dots$

The integers satisfy the nat fixed-point construction axiom. But from the other given formula we know $0 \leq n$ for each element n , so that rules out all the negative numbers. Maybe nat includes ∞ .

$0, 1, 2, 3, \dots, \infty$

That satisfies the nat fixed-point construction axiom. But from the other given formula we know $n < n+1$ for each element n , so that rules out ∞ . Maybe there are elements in between the natural numbers, like this:

$0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, \dots$

This bunch satisfies ordinary nat construction, but not fixed-point construction. If this were nat , then $0, \text{nat}+1$ would be

$0, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, \dots$

missing 0.5 . Similarly for any other elements in between the natural numbers. So nat must be the natural numbers. Most mathematicians consider this to be a proof, but I don't. I want formal proof, but I don't know how to prove it formally.

(b) $\exists m: \text{nat} \cdot \forall n: \text{nat} \cdot m \leq n < n+1$

§ This is like part (a), but instead of $0 \leq n$ we have $m \leq n$ for some m . This makes it harder to rule out the negative integers.