



A proof of $\perp$. The environment Δ has the natural numbers as three primitive definitions, and induction as a fourth:

$$\begin{aligned} \mathbb{N} &:= \perp\!\!\!\perp : *_s & 0 &:= \perp\!\!\!\perp : \mathbb{N} & s &:= \perp\!\!\!\perp : \mathbb{N} \rightarrow \mathbb{N} \\ \text{ind} &:= \perp\!\!\!\perp : \prod P : \mathbb{N} \rightarrow *_p. (\forall n : \mathbb{N}. (P n \Rightarrow P (s n))) \Rightarrow \forall n : \mathbb{N}. P n \end{aligned}$$

In it, with the empty context, $\perp$ has a proof. Every line follows by a rule, and the kernel accepts it:

- | | | |
|-----|---|---|
| (1) | $a_1 := \lambda n : \mathbb{N}. \lambda u : \perp. u : \forall n : \mathbb{N}. (\perp \Rightarrow \perp)$ | $\forall$ -in, $\Rightarrow$ -in |
| (2) | $a_2 := \text{ind}(\lambda n : \mathbb{N}. \perp) a_1 : \forall n : \mathbb{N}. \perp$ | Π -el, (<i>conv</i>), $\Rightarrow$ -el |
| (3) | $a_3 := a_2 0 : \perp$ | $\forall$ -el on 0 |

The same in Lean, with $n + 1$ for sn ; the audit lists *ind* only.

```
axiom ind : ∀ P : Nat → Prop, (∀ n, P n → P (n + 1)) → ∀ n, P n
theorem wrong : False := ind (fun _ => False) (fun _ u => u) 0
#print axioms wrong
```

lines (1)–(3)
[ind]

- (a) $\perp$ has no proof, yet line (3) is one and the kernel accepts it. Why? Where exactly is the mistake?
- (b) Correct it. Which line of (1)–(3) then breaks, and what would you have to supply there?

After Nederpelt & Geuvers, *Type Theory and Formal Proof* (CUP, 2014), Exercise 10.2(c). Lean: *Exercises.lean*, section 11.

I Definitions

- **Primitive definition.** $a := \perp\!\!\!\perp : N$ adds a constant with no body; it cannot be unfolded. The kernel checks that N is a well-formed type. An audit lists the primitive constants left after unfolding every descriptive one.
- Π -el, $\forall$ -el: from $u : \prod x : S. P x$ and $a : S$, $u a : P a$. $\Rightarrow$ -el likewise.
- $\forall$ -in: from $M : P x$ under a flag $x : S$, $\lambda x : S. M : \forall x : S. P x$, closing the flag. $\Rightarrow$ -in likewise.
- (*conv*): a term of type B may be used at any $B' =_{\beta\delta} B$. $\perp := \prod \alpha : *_p. \alpha$.

Name _____

Student ID _____

Answer

Out of room? Continue on the back of the page.