

Lean learning group

What?

$\therefore \sim(P \rightarrow Q) \rightarrow (P \wedge \sim Q)$

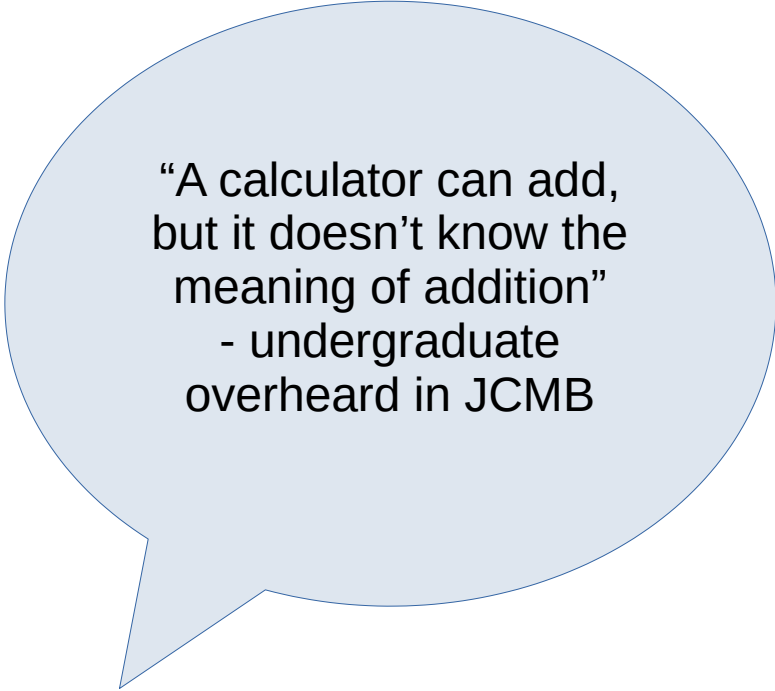
1.	show $\sim(P \rightarrow Q) \rightarrow (P \wedge \sim Q)$	
2.	$\sim(P \rightarrow Q)$	ass cd
3.	show $P \wedge \sim Q$	
4.	show P	
5.	$\sim P$	ass id
6.	show $P \rightarrow Q$	
7.	P	ass cd
8.	$\sim P$	5, r
9.		7, 8, id
10.	$\sim(P \rightarrow Q)$	2, r
11.		6, 10, id
12.	show $\sim Q$	
13.	Q	ass id
14.	show $P \rightarrow Q$	
15.	Q	13, r
16.		15, cd
17.	$\sim(P \rightarrow Q)$	2, r
18.		14, 17, id
19.	$P \wedge \sim Q$	4, 12, adj
20.		19, dd
21.		3, cd

```

lemma lt_aux_one (a b : mynat) : a ≤ b ∧ ¬ (b ≤ a) → succ a ≤ b :=
begin [nat_num_game]
  intro h,
  cases h with h1 h2,
  cases h1 with c hc,
  cases c with d,
    exfalse,
    rw add_zero at hc,
    apply h2,
    rw hc,
    refl,
  use d,
  rw hc,
  rw add_succ,
  rw succ_add,
  refl,
end

```

Why?



“A calculator can add,
but it doesn’t know the
meaning of addition”
- undergraduate
overheard in JCMB

- Is ChatGPT going to replace us?
- Or help us? A future proof assistant in your pocket?
- Get a postdoc?

About this course

- Learning outcomes:
 - Learn the basics of formalising mathematics in Lean
 - Learn to love interacting with your computer
 - Learn some general programming skills (paired programming, git...)
- Learning non-outcomes:
 - More technical CS aspects of Lean, type theory etc
 - Learning an imperative language
 - School of Mathematics Python Programming course in Semester 1 (UoE: equivalents in Gla/HW?)
 - Self-paced notebooks: <https://github.com/PatrickKinnear/python-intro>
 - Wealth of resources available online

About this course

- We are not experts! Spirit of the course is co-creation.
- Three broad sections:
 - Basics of Lean
 - Exploring mathlib
 - Project: formalising a new theorem?
- Course info: on the website and Zulip stream
 - <https://www.maths.ed.ac.uk/~pkinnear/leancourse>
- Today: natural number game