| Prosemino | r on computer-assisted mal | hematics |
|---|---|---|
| · · · · · · · · · · · · · · · · · · · | ession 7 – Introduction to Leav | |
| . . | | |
| . . | THEOREM PROVER | |
| Heidelbe | Florent Schaffhauser erg University, Summer semester | $\frac{2023}{1000000000000000000000000000000000000$ |

| · · · | Today we will: | | | | · · · · · · · · · · · · · · · · · · · |
|---------------------------------------|---|---------------------------------------|--|---|---|
| · · · · · · · · · · · · · · · · · · · | · · · · · · · · · · · · · · · | · · · · · · · · · · · · · · · · · · · | | Learn about Lean Theorem Prover, a programming language and proof assistant created by Leonardo de Moura in 2013. | . .< |
| • • | | | | | |
| · · | · · · | · · · | | Get acquainted with Lean 3 and prove our first propositions, which will all be equalities between objects. | · · · · |
| · · · · · · · · · · · · · · · · · · · | · · · · · · · · · · · · · · | · · · | | Get acquainted with Lean 3 and prove our first propositions, which will all be equalities between objects. | · · · · · · · · · · · · · · · · · · · |
| · · · · · · · · · · · · · · · · · · · | . . | · · · · · · · · · · · · · · · · · · · | | Get acquainted with Lean 3 and prove our first propositions, which will all be equalities between objects. | . .< |
| | . . | · · · · · · · · · · · · · · · · · · · | | Get acquainted with Lean 3 and prove our first propositions, which will all be equalities between objects. | . .< |

| Resources about Lean | |
|---|---|
| We will be using Lean 3. started is the Lean comm | The best place to get runity website: |
| Image: state of the state o | <u>https://leanprover-</u> <u>community.github.io/</u> |
| In particular the installation instructions. | https://leanprover- community.github.io/ get_started.html |
| | |

| In a web browser: <u>Lean-web-editor</u> <u>Lean-web-editor</u> <u>Load lean from URL:</u> Load lean from URL: Load lean from the: <u>Consected</u> <u>Load lean from the Prop</u> : PA (P = 0) = 0 := | In a web browser: Community.github.io/ Lean-web-editor Lean-web-editor Load lean from URL: Load lean from URL: Load lean from disk: Choose File not fir selected I the in howser version if the Lean Herren prove i the in howser version if the Lean Herren prove i the herren prove the cursor in the Lean file and read the response from the program on the right. | | •••• | | | | | https://L | eanbri | over. | | · · · · |
|---|--|---------|------|-----|------|--------------------------------|---|--|----------|---------|------|---------|
| Lean is ready Load lean from URL: Load lean from disk: <u>Chose File</u> no file selected Load lean from disk: <u>Chose File</u> no file selected The Lean house of the Lean house muran. | Lean is ready Load Jean from URL: Load Jean from URL: Load Jean from URL: Load Jean from URL: Load Jean from URL: Load Jean from URL: Inter Lean Jean from URL: Inter Jean 2 /- ***flotus ponens*** -/ St: goal 2 /- ***flotus ponens*** -/ St: goal 3 d def MP (P 0 : Prop) : P A (P + 0) + 0 := St: goal Cases h with MP NPQ, Stift of MP (P 0 : Prop) : P A (P + 0) + 0 := 3 d def MP (P 0 : Prop) : P A (P + 0) + 0 := St: goal 1 cases h with MP NPQ, Stift of MP (P 0 : Prop) : P A (P + 0) + 0 := 3 d def MP (P 0 : Prop) : P A (P + 0) + 0 := St: goal 1 end The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | | II | n a | web brow | ser: | commu | ily.gi | thub. | io/ | , , |
| Usage: move the cursor in the Lean file and read the response from the program on the right. | Usage: move the cursor in the Lean file and read the response from the program on the right. | | • • | • • | · · | · · · · · · · · · | · · · · · · · · · · · | Lean-we | b-edil | or | | |
| Load lean from URL: Load lean from disk: Choose File no file selected Live in-browser version of the Lean theorem prover. $\frac{1}{2} / - = the forward version of the Lean theorem prover.$ $\frac{1}{2} / - = the forward version of the Lean theorem prover.$ $\frac{1}{2} / - = the forward version of the Lean theorem prover.$ $\frac{1}{2} / - = the forward version of the Lean theorem prover.$ $\frac{1}{2} / - = the forward version of the Lean theorem prover.$ $\frac{1}{2} / - = the forward version of the Lean theorem prover.$ $\frac{1}{2} / - = the forward version of the Lean theorem prover.$ $\frac{1}{2} / - = the forward version of the lean theorem prover.$ $\frac{1}{2} / - = the forward version of the lean theorem prover.$ The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | Load lean from URL: Load lean from disk: Choose File to file selected 2 Live in-browser version of the Lean theorem prover. | | • • | • • | • • | | ✓ Lean is ready! | | • • • | | • • | • • • |
| Usage: move the cursor in the Lean file and read the response from the program on the right. | Usage: move the cursor in the Lean file and read the response from the program on the right. | • • • • | • • | • • | • • | | Load lean from LIRI · | | Load | | • • | |
| Image: Intro b, Image: Ima | Use the inhower version of the Lean theorem prover. Save 2 /- ***Nodus ponens*** -/ 3 4 def MP (P 0 : Prop) : P x (P - 0) - 0 := 5 begin intro h, 7 (cases h with nP hP0, 8 oppive watt hP, 9 exact hP, 10 end 5 begin intro h, 7 the Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | | | | | Load .lean from disk: Choose File no file | selected | Load | | | |
| I def MP (P 0 : Prop) : P A (P - 0) - 0 := I def MP (P 0 : P A (P - 0) - 0 := I def MP (P 0 : P A (P - 0) - 0 := I def MP (P 0 : P A (P - 0) - 0 := I def MP (P 0 : P A (P - 0) - 0 := I def MP (P 0 : P A (P - 0) - 0 := I def MP (P 0 : P A (P - 0) - 0 := I def MP (P 0 : P A (P - 0) - 0 := I def MP (P 0 : P A (P - 0) - 0 := I def M | 1 2 /- ***Modus ponens*** -/ 3 def MP (P 0 : Prop) : P A (P + 0) - 0 := begin 4 def MP (P 0 : Prop) : P A (P + 0) - 0 := P 0 : Prop 5 begin point intro h, 6 intro h, cases h with hP hP0, 9 exact hP, point interface 10 end The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | | | | | ? Live in-browser version of the Lean theorem | <u>ı prover</u> . | Save | | | |
| $\frac{2}{2} / - ***Modus ponens** - / \\ \frac{2}{3} def MP (P Q : Prop) : P A (P - 0) + Q := \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) + Q & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) + Q & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) + Q & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) + Q & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) + Q & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) & \\ \frac{2}{5} begin{trace}{l} F P A (P - 0) & \\ \frac{2}{5} begin{trac$ | $\frac{2}{4} = \frac{1}{4} + \frac{1}$ | | • • | • • | • • | · 1 | | 5:5: doal | | • • • • | | |
| $\begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} & & \\ & $ | $\int_{0}^{3} \frac{def MP (P Q : Prop) : P A (P + Q) + Q :=}{\int_{0}^{2} \frac{P Q : Prop}{P A (P + Q) + Q}}$ $\int_{0}^{2} \frac{1}{p P A (P + Q) + Q}$ $\int_{0}^{2} \frac{1}{p P A (P + Q) + Q}$ $\int_{0}^{2} \frac{P Q : Prop}{P A (P + Q) + Q}$ $\int_{0}^{2} P Q (P + Q) +$ | | | | | 2 /- ***Modus ponens*** -, | / | | | | | |
| S begin G intro h, cases h with hP hPQ, apply hPQ, g exact hP, 10 end The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | 5 beain 6 intro h, cases h with PP PQ, apply hPQ, 9 end 10 end The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | • • | | | 3 4 def MP {P Q : Prop} : P | $\Lambda (P \rightarrow Q) \rightarrow Q :=$ | $\vdash P \land (P \rightarrow Q) \rightarrow Q$ | | | | |
| Usage: move the cursor in the Lean file and read the response from the program on the right. | The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | | • • | • • | 5 begin | | | | | | • • • |
| Usage: move the cursor in the Lean file and read the response from the program on the right. | Usage: move the cursor in the Lean file and read the response from the program on the right. | | • • | | • • | 7 cases h with hP hPQ, | | | | | • • | |
| The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | 0 0 | • • | • • | 8 apply hPQ, 9 exact hP. | | | | | | |
| The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | • • | • • | | 10 end | | | | , | | • • • |
| The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | | | | • | | | | | | |
| The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | • • | • • | • • | | | | | | • • | |
| The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right, | The Lean web editor interface Usage: move the cursor in the Lean file and read the response from the program on the right. | | | • • | • • | | | | | | • • | • • • |
| Usage: move the cursor in the Lean file and read the response from the program on the right. | Usage: move the cursor in the Lean file and read the response from the program on the right. | | • • | • • | • • | The L | ean web edito | r interface | | | • • | |
| Usage: move the cursor in the Lean file and read the response from the program on the right. | Usage: move the cursor in the Lean file and read the response from the program on the right. | | | • • | | | | · · · · · · · · · · · · · · · · · · · | • • • • | | | |
| Usage: move the cursor in the Lean file and read the response from the program on the right. | Usage: move the cursor in the Lean file and read the response from the program on the right. | | • • | • • | • • | | | | | | | |
| Usage: move the cursor in the Lean file and read the response from the program on the right. | Usage: move the cursor in the Lean file and read the response from the program on the right. | | • • | • • | • • | | | | | | | • • • |
| the response from the program on the right. | the response from the program on the right. | | • • | • • | 1 | SADE MADVA | the cursor in 1 | he Loon fil | eand | read | • • | • • • |
| the response from the program on the right. | the response from the program on the right. | • • • • | | • • | . V | NOTE FROM | | FOR MELAPA JUL | s wright | I SUM | • • | |
| n an Anna an Africana an an an Anna Merican Mericana an an an Africana an an an a Anna an | | • • • • | | • • | - El | ne response | from the prog | ram on the | right. | 1 0 0 0 | • • | |
| | | • • • • | • • | | • • | | a de la companya de l | | « | | • • | • • • |

| n an | na na sana na Mana na Rana na sana Rate na sana na Rana na Rana na Rana na sana na sana tena tena tena tena tena tena na sana |
|--|--|
| For us, the most convente | enc option today will be |
| to upload our Lean file t | o CoCalc: " " " " " " " " " " " " " |
| | |
| | |
| | |
| · · · · · Input | ① Info at Cursor ① \/ - + ··· ① Info \/ _ _ : [] [] [] X · · Tactic State |
| $\frac{1}{1}$ | PQ: Prop, 7 What is shown here will |
| $\frac{1}{3} \det MP \{P Q : Prop\} (hP : P) (hPQ : P \rightarrow Q) : Q :=$ | hP: P, hPQ: P + Q depend on where the |
| 5 begin 6 apply hPQ, | register is in the file. |
| 7 exact hP, • 8 end | |
| · · · 10 () #check @MP | ⊕ Help at Cursor ⊕ Help x ⊕ Help x ⊕ Help x |
| 12 /- MP appears as a function that, given propositions P and Q, sends a proof of the propositions P and $(P \rightarrow Q)$ to a proof of Q | |
| 13 14 MP : \forall {P Q : Prop}, P \rightarrow (P \rightarrow Q) \rightarrow Q | |
| 15 -/ 16 17 warishing (D.O., Prep) (bp., D. (bp., D., O) | |
| 18 19 /- A proof that, in our context, the Proposition 0 is true: we | · · · · |
| simply apply the *modus ponens* function defined above to the proofs of the propositions P and $(P \rightarrow Q) -/$ | |
| 20 21 def In_our_context_Q_is_true : Q := MP hP hPQ | |
| 22 23 () #check @MP P Q hP hPQ 24 () #check MP hP hPQ | ① 10:0 information check result |
| | $MP : V \{P Q : Prop\}, P \rightarrow (P \rightarrow Q) \rightarrow Q$ |
| | |
| | The doto shared by files and |
| The Lean file | The data stored in the |
| | object called Mr, defined |
| | between Lines 4 and 8. |
| | ne en la companya de la companya de La companya de la comp |
| | |

Now we practice!

The first practice file is the Introduction to Lean file, available from the seminar webpage.

| ≅ week1.lean Lean/week1.lean/ | J 🖬 🛙 |) × … |
|---|--|--|
| /- # **Introduction to Lean** | 1.4 Section of the test minimum section of the test minimum section of the test minimum section of the test minimum section of the test minimum section of the test minimum section of the test minimum section of the test minimum section of the test minimum section of test minimum section of the test minimum section of test | a) Solutioning have based All and the contrast of a series and an end of a series of a series of a series of a series of a series of a series of a series of a series based of a series of a series of a series of a series based of a series of a series of a series of a series based of a series |
| **Author:** Florent Schaffhauser, Uni-Heidelberg, Summer Semester 2023 | | |
| **Lean is a programming language** that can be used as a *proof assistant | * (also | |
| This means that Lean can be used to check and certify the correctness of | certair | F) - Max, and P and same, at an an array of an array of the same and an array of the same array bases of an array of the same array. |
| It was created and first implemented by **Leonardo de Moura ** at Microsof | t Reset | $M_{\rm eff}^{\rm eff}$ gives the state of the |
| The current version is Lean 4, dating back to 2021. It is not backwards-co | ompatil | And Teleford (Section 201) Context Co |
| /- ## Types and terms -/ | Management of an antiparticle of a state of the state of | |
| /- In Lean, we have access to certain data types, which are part of the ${\tt L}$ | anguage | Solar |
| The command `#check` tells us the type of an expression, for instance `ch | ar` foi | per de la fait a serie en ante des las des las des las estes en ante des las des las des las estes en ante de las de las de las des las des las de las de las de las de las de las de las de personantes, en destantes de la de las de las de las de personantes destantes de las de las de las de las de las de las de las de las d |
| If `#check t` returns `T`, one says that **`t` is a term of type `T`** . T | his is | In the control control, which is not the start is "they will be start at a part control control of the start of the start and part control before the start control of the start and part control before the start part of the start and part of the start part of the start and part of the start part of the start of the start part of the start part of the start of the start part of the start part of the start of the start par |
| <pre>#check 'H' 'H' : char #check "Hello, world!" "Hello, world!" : string #check 42 42 : N</pre> | In In< | $\label{eq:second} \begin{array}{l} 0 & \mathrm{prod} & \mathrm{density} & den$ |
| Eean Infoview | 1 10 10 of 10 at 10 | × … |
| ▼week1.lean:9:0 | י וו ⊭- |) 💬 |
| No info found. | | |
| ► All Messages (35) | | П |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Preview week1.md

Introduction to Lean

Author: Florent Schaffhauser, Uni-Heidelberg, Summer Semester 2023

Lean is a programming language that can be used as a *proof assistant* (also called an *interactive theorem prover*).

This means that Lean can be used to check and certify the correctness of certain computer programmes and formalised mathematical proofs.

It was created and first implemented by **Leonardo de Moura** at Microsoft Research, where the first version was launched in 2013.

The current version is Lean 4, dating back to 2021. It is not backwards-compatible wih Lean 3, which is the version that we use for the purposes of this seminar.

Types and terms

In Lean, we have access to certain data types, which are part of the language.

The command **#check** tells us the type of an expression, for instance **char** for a character, **string** for a string of characters, and N (also called **nat**) for an integer. This last one will turn out to be of a different "nature" than the first two.

If #check t returns T, one says that t is a term of type T. This is abbreviated to t : T.

#check 'H'
#check "Hello, world!"
#check 42

#check "Hello, ".append("world!")
#check 41 + 1

The data types string and \mathbb{N} are themselves terms of type Type. You can obtain the symbol \mathbb{N} by typing \nat or \N followed by the space bar. You can also just use nat instead of \mathbb{N} .

#check string
#check ℕ
#check nat

Not all data types are terms of type Type. Some are more complex than that, for instance the type list.

On the Github repository of the seminar, you can find that file under the name week1.lean, along with its markdown version.

| Provide Comp_assisted_math / Lean / | | Q Go to file t |
|---|---|------------------------------------|
| matematiflo Practice folder created | | e77acef · 10 minutes ago 🕚 History |
| Name | Last commit message | Last commit date |
| 🖿 | | |
| Practice_folder | Practice folder created | 10 minutes ago |
| C README.md | Update README.md | 1 hour ago |
| 🗋 week1.lean | Updated week1 files | 3 hours ago |
| week1.md | Updated week1 files | 3 hours ago |
| README.md | | P |
| Formal proofs in Lean 3 | | |
| Author: Florent Schaffhauser, Uni-Heidelberg, Summ | er Semester 2023 | |
| This is the GitHub repository for the <i>Introduction to Le</i> University of Heidelberg in the Summer Semester of 2 | ean part of the (Pro)Seminar on computer- 2023. | assisted mathematics, held at the |

Below you will find the programme of the seminar. For each week, there is a corresponding .lean file, which you can use to practice. You can also view a markdown version of the weekly files by clicking on the corresponding .md files.

I have also put the modus ponens file in the Practice folder.