The Natural Number Game : an introduction to Lean tactics

K. Buzzard

13th July 2020, LFTCM2020

Main learning objectives for this afternoon's session:

Main learning objectives for this afternoon's session:

- You will understand the `refl`, `rw` and `induction` tactics.

Main learning objectives for this afternoon's session:

- You will understand the `refl`, `rw` and `induction` tactics.
- You will be aware of the `intros`, `exact`, `apply`, `use`, `split` and `cases` tactics.

Main learning objectives for this afternoon's session:

- You will understand the `refl`, `rw` and `induction` tactics.
- You will be aware of the `intros`, `exact`, `apply`, `use`, `split` and `cases` tactics.
- You will get used to writing basic proofs in Lean.

Main learning objectives for this afternoon's session:

- You will understand the `refl`, `rw` and `induction` tactics.
- You will be aware of the `intros`, `exact`, `apply`, `use`, `split` and `cases` tactics.
- You will get used to writing basic proofs in Lean.
- You will have Lean and `leanproject` installed correctly.

Main learning objectives for this afternoon's session:

- You will understand the `refl`, `rw` and `induction` tactics.
- You will be aware of the `intros`, `exact`, `apply`, `use`, `split` and `cases` tactics.
- You will get used to writing basic proofs in Lean.
- You will have Lean and `leanproject` installed correctly.

Note: if you know all this stuff already, Rob Lewis is running a parallel thread on metaprogramming (i.e. tactic-writing).

Brief introduction

I initially got interested in Lean because I thought it might be a cool teaching tool.

Brief introduction

I initially got interested in Lean because I thought it might be a cool teaching tool.

After trying out some material with undergraduates, I found that they enjoyed proving lemmas about natural numbers from first principles.

Brief introduction

I initially got interested in Lean because I thought it might be a cool teaching tool.

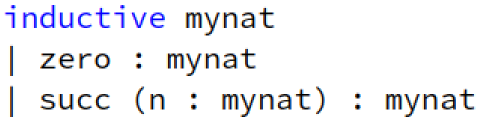
After trying out some material with undergraduates, I found that they enjoyed proving lemmas about natural numbers from first principles.

I made the natural number game with Mohammad Pedramfar.

Here is what is going on behind the scenes (this is optional material).

Here is what is going on behind the scenes (this is optional material).

The type `mynat` is defined in Lean as an *inductive type*.



```
inductive mynat
| zero : mynat
| succ (n : mynat) : mynat
```

Here is what is going on behind the scenes (this is optional material).

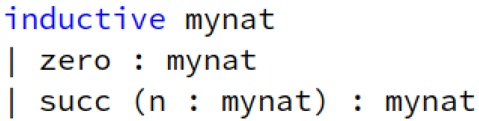
The type `mynat` is defined in Lean as an *inductive type*.

```
inductive mynat
| zero : mynat
| succ (n : mynat) : mynat
```

This Lean code creates four things:

Here is what is going on behind the scenes (this is optional material).

The type `mynat` is defined in Lean as an *inductive type*.



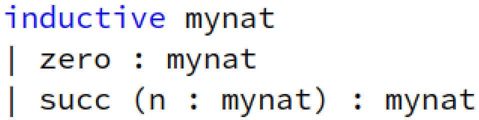
```
inductive mynat
| zero : mynat
| succ (n : mynat) : mynat
```

This Lean code creates four things:

- A new type called `mynat`;

Here is what is going on behind the scenes (this is optional material).

The type `mynat` is defined in Lean as an *inductive type*.



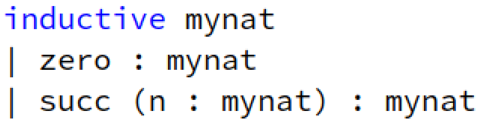
```
inductive mynat
| zero : mynat
| succ (n : mynat) : mynat
```

This Lean code creates four things:

- A new type called `mynat`;
- New terms `mynat.zero` and `mynat.succ`;

Here is what is going on behind the scenes (this is optional material).

The type `mynat` is defined in Lean as an *inductive type*.



```
inductive mynat
| zero : mynat
| succ (n : mynat) : mynat
```

This Lean code creates four things:

- A new type called `mynat`;
- New terms `mynat.zero` and `mynat.succ`;
- A recursor `mynat.rec`.

The recursor is hidden from the user in the natural number game.

The recursor is hidden from the user in the natural number game.

The recursor is a function which enables you to define functions by recursion on the naturals, *and* prove things by induction.

The recursor is hidden from the user in the natural number game.

The recursor is a function which enables you to define functions by recursion on the naturals, *and* prove things by induction.

This can be done, because type/term generalises element/set and proposition/proof.

The recursor is hidden from the user in the natural number game.

The recursor is a function which enables you to define functions by recursion on the naturals, *and* prove things by induction.

This can be done, because type/term generalises element/set and proposition/proof.

Definitions (for example addition of natural numbers) use the recursor and are hidden from the user.

The recursor is hidden from the user in the natural number game.

The recursor is a function which enables you to define functions by recursion on the naturals, *and* prove things by induction.

This can be done, because type/term generalises element/set and proposition/proof.

Definitions (for example addition of natural numbers) use the recursor and are hidden from the user. They are written in term mode.

The recursor is hidden from the user in the natural number game.

The recursor is a function which enables you to define functions by recursion on the naturals, *and* prove things by induction.

This can be done, because type/term generalises element/set and proposition/proof.

Definitions (for example addition of natural numbers) use the recursor and are hidden from the user. They are written in term mode.

All proofs are written in tactic mode (within a `begin`/`end` block).

OK so let's get started! All links are in the "Lean for the curious mathematician" stream, in the "Mon Afternoon: natural number game" topic.

OK so let’s get started! All links are in the “Lean for the curious mathematician” stream, in the “Mon Afternoon: natural number game” topic.

By the end of this session, you should be able to do Tutorial World -> Addition World -> Multiplication World -> Power World (`induction`, `rw`, `refl`).

OK so let's get started! All links are in the "Lean for the curious mathematician" stream, in the "Mon Afternoon: natural number game" topic.

By the end of this session, you should be able to do Tutorial World -> Addition World -> Multiplication World -> Power World (`induction`, `rw`, `refl`). If you want to try and get to inequality world, that's great.

OK so let's get started! All links are in the "Lean for the curious mathematician" stream, in the "Mon Afternoon: natural number game" topic.

By the end of this session, you should be able to do Tutorial World -> Addition World -> Multiplication World -> Power World (`induction`, `rw`, `refl`). If you want to try and get to inequality world, that's great.

If you have finished the natural number game, then try the **lost levels.**

OK so let's get started! All links are in the "Lean for the curious mathematician" stream, in the "Mon Afternoon: natural number game" topic.

By the end of this session, you should be able to do Tutorial World -> Addition World -> Multiplication World -> Power World (`induction`, `rw`, `refl`). If you want to try and get to inequality world, that's great.

If you have finished the natural number game, then try the **lost levels.** Install the natural number game locally using `leanproject`, open the project (open *folder*), open the file `src/game/world10/level18q.lean`, and fill in the `sorry`s!

OK so let's get started! All links are in the "Lean for the curious mathematician" stream, in the "Mon Afternoon: natural number game" topic.

By the end of this session, you should be able to do Tutorial World -> Addition World -> Multiplication World -> Power World (`induction`, `rw`, `refl`). If you want to try and get to inequality world, that's great.

If you have finished the natural number game, then try the **lost levels.** Install the natural number game locally using `leanproject`, open the project (open *folder*), open the file `src/game/world10/level18q.lean`, and fill in the `sorry`s!

If you don't understand anything – ask! Zoom, Zulip etc.