Injections

Let X and Y be sets. In the last video : counting all the functions from X to Y.

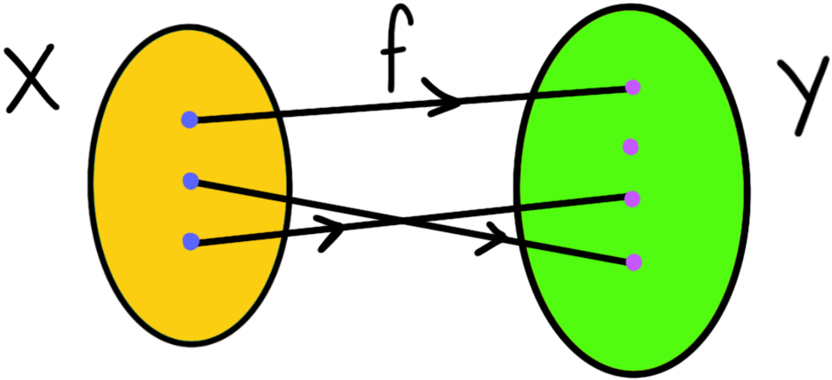
But here are some predicates on the set $\mathrm{Hom}(X, Y)$ which make things more interesting.

In the remaining "functions" videos, we will be looking at three special kinds of functions called *injections*, *surjections* and *bijections*.

These are the nouns – the corresponding adjectives are *injective* functions, *surjective* functions and *bijective* functions.

In this video we will talk about injective functions.

A function $f : X \to Y$ is *injective* if it sends distinct elements of X to distinct elements of Y. Let's start with a picture.



A function $f : X \to Y$ is *injective* if it sends distinct elements of X to distinct elements of Y.

Non-example: $f(x) = x^2$ is a function from $\mathbb{R}$ to $\mathbb{R}$, but it is *not* injective, because $f(2) = 4$ and $f(-2) = 4$.

Put it another way: a function is *not injective* if there exists $a, b \in X$ with $a \neq b$ but $f(a) = f(b)$.

Put it another way: a function f is injective if $\forall a, b \in X, f(a) = f(b) \implies a = b$.

This last definition is the preferred definition in a test or exam.

Maybe some of you learnt the phrase “one-to-one”, or “into”, to describe such functions.

At university we will call them “injective” functions.

Here is yet another way of thinking about this idea.

Say $f : X \to Y$ is a function, and say $y \in Y$.

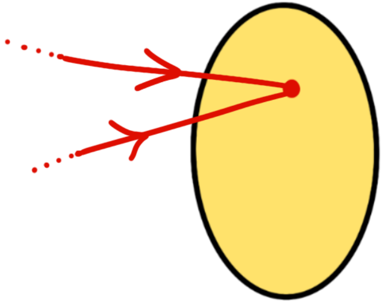
New notation: The *pre-image* of y is the subset of X consisting of all the elements of X which are sent to y by f.

In other words, the preimage of y is $\{x \in X \mid f(x) = y\}$.

Example: the preimage of 4 under the squaring function from $\mathbb{R}$ to $\mathbb{R}$ is the set $\{2, -2\}$.

Another way of picturing injectivity: the function f is injective if and only if the preimage of any element of Y has *size at most 1*.

Sole obstruction to injectivity



The preimage of the point in Y is at least two points in X.

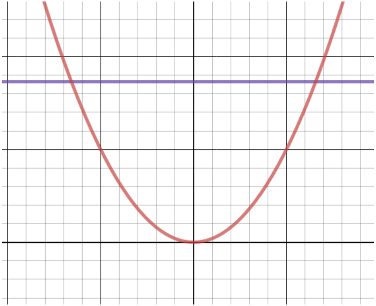
Here is yet another way of thinking about injectivity.

If $y \in Y$ then let's define the *horizontal line* through y to be the subset $X \times \{y\}$.

A function $f : X \to Y$ is injective if, and only if, it satisfies the **Horizontal line test for injectivity:** Every horizontal line meets the graph of f in *at most one point*.

Here is a non-example.

The function $x \mapsto x^2$ from $\mathbb{R}$ to $\mathbb{R}$ is not injective, because it fails the horizontal line test: the blue line does *not* meet the graph of f in *at most one point*.



Get a geometric intuition for injectivity. But remember $\forall a, b \in X, f(a) = f(b) \implies a = b$.

Let's now prove some theorems about injectivity.

Because we are proving theorems, we will use the recommended definition:

$f : X \to Y$ is injective if and only if

$$\forall a, b \in X, f(a) = f(b) \implies a = b.$$

First question. Is the identity function injective? (think about the horizontal line test for injectivity).

Theorem

Let X be a set. Then the identity function $id_X : X \to X$ is injective.

Why don't you pause the video and have a go at proving this yourself? Then I will show you my proof.

Proof.

We are asked to prove that
$\forall a, b \in X, id_X(a) = id_X(b) \implies a = b$.

So, let a and b be arbitrary elements of X.

We have to prove $id_X(a) = id_X(b) \implies a = b$.

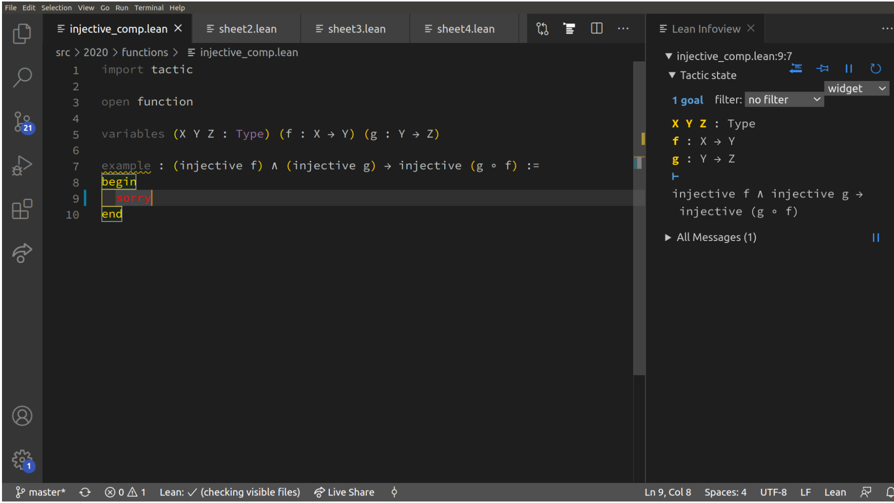
But by definition of id_X, our job is to prove $a = b \implies a = b$.

This is true! So the proof is over. $\square$

Theorem

Let X, Y and Z be sets, and let $f : X \to Y$ and $g : Y \to Z$ be functions.
If f and g are both injective, then $g \circ f$ is also injective.

The proof of this is in the course notes. Let me show you what the proof looks like in Lean, and if you don't care about Lean then you can just read the course notes.


File Edit Selection View Go Run Terminal Help
injective_comp.lean
sheet2.lean
sheet3.lean
sheet4.lean
src > 2020 > functions > injective_comp.lean
import tactic
open function
variables (X Y Z : Type) (f : X → Y) (g : Y → Z)
example : (injective f) ∧ (injective g) → injective (g ∘ f) :=
begin
sorry
end
Lean Infoview
injective_comp.lean:9:7
Tactic state
widget
1 goal
filter: no filter
X Y Z : Type
f : X → Y
g : Y → Z
⊢
injective f ∧ injective g →
injective (g ∘ f)
All Messages (1)
master*
0 1
Lean: ✓ (checking visible files)
Live Share
Ln 9, Col 8
Spaces: 4
UTF-8
LF
Lean

File Edit Selection View Go Run Terminal Help

injective_comp.lean | sheet2.lean | sheet3.lean | sheet4.lean

src > 2020 > functions > injective_comp.lean

```lean
import tactic

open function

variables (X Y Z : Type) (f : X → Y) (g : Y → Z)

example : (injective f) ∧ (injective g) → injective (g ∘ f) :=
begin
  -- Assume that f and g are injective.

  -- We need to prove `g ∘ f` is injective.
  -- So say `a`, `b` are in `X`, and `(g ∘ f)(a) = (g ∘ f)(b)`

  -- We need to prove that `a = b`.
  -- By injectivity of `f`, it suffices to prove that `f(a) = f(b)`.

  -- But this follows immediately from our assumption
  -- `g(f(a))=g(f(b))`,and injectivity.

end
```

Lean Infoview

▼ injective_comp.lean:20:0

▼ Tactic state

widget

```
expected type:
X Y Z : Type
f : X → Y
g : Y → Z
⊢
injective f ∧ injective g →
  injective (g ∘ f)
```

▼ Messages (1)

▼ injective_comp.lean:20:0

```
tactic failed, there are
unsolved goals
state:
X Y Z : Type,
f : X → Y,
g : Y → Z
⊢ injective f ∧ injective g →
injective (g ∘ f)
```

▶ All Messages (1)

master* ⊗ 1 ⚠ 0 Lean: ✓ (checking visible files) Live Share Blame Kevin Buzzard (3 days ago) Ln 20, Col 1 Spaces: 4 UTF-8 LF Lean

File Edit Selection View Go Run Terminal Help

injective_comp.lean | sheet2.lean | sheet3.lean | sheet4.lean

src > 2020 > functions > injective_comp.lean

```
import tactic

open function

variables (X Y Z : Type) (f : X → Y) (g : Y → Z)

example : (injective f) ∧ (injective g) → injective (g ∘ f) :=
begin
  -- Assume that f and g are injective.
  rintro (hf, hg),
  -- We need to prove `g ∘ f` is injective.
  -- So say `a`, `b` are in `X`, and `(g ∘ f)(a) = (g ∘ f)(b)`
  sorry
  -- We need to prove that `a = b`.
  -- By injectivity of `f`, it suffices to prove that `f(a) = f(b)`.

  -- But this follows immediately from our assumption
  -- `g(f(a))=g(f(b))`,and injectivity.

end
```

Lean Infoview

▼ injective_comp.lean:10:18

▼ Tactic state

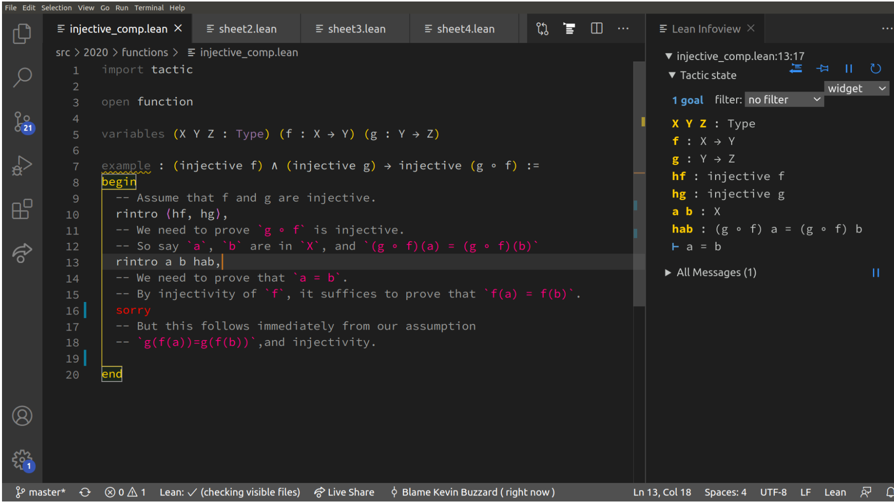
widget

1 goal filter: no filter

```
X Y Z : Type
f : X → Y
g : Y → Z
hf : injective f
hg : injective g
⊢ injective (g ∘ f)
```

▶ All Messages (1)

master* ⊗ 0 ⚠ 1 Lean: ✓ (checking visible files) Live Share Blame Kevin Buzzard (right now) Ln 10, Col 19 Spaces: 4 UTF-8 LF Lean


File Edit Selection View Go Run Terminal Help
injective_comp.lean
sheet2.lean
sheet3.lean
sheet4.lean
src > 2020 > functions > injective_comp.lean
import tactic
open function
variables (X Y Z : Type) (f : X → Y) (g : Y → Z)
example : (injective f) ∧ (injective g) → injective (g ∘ f) :=
begin
-- Assume that f and g are injective.
rintro ⟨hf, hg⟩,
-- We need to prove `g ∘ f` is injective.
-- So say `a`, `b` are in `X`, and `(g ∘ f)(a) = (g ∘ f)(b)`
rintro a b hab,
-- We need to prove that `a = b`.
-- By injectivity of `f`, it suffices to prove that `f(a) = f(b)`.
sorry
-- But this follows immediately from our assumption
-- `g(f(a))=g(f(b))`,and injectivity.
end
Lean Infoview
injective_comp.lean:13:17
Tactic state
widget
1 goal filter: no filter
X Y Z : Type
f : X → Y
g : Y → Z
hf : injective f
hg : injective g
a b : X
hab : (g ∘ f) a = (g ∘ f) b
⊢ a = b
All Messages (1)
master* 0 1 Lean: ✓ (checking visible files) Live Share Blame Kevin Buzzard (right now) Ln 13, Col 18 Spaces: 4 UTF-8 LF Lean

File Edit Selection View Go Run Terminal Help

injective_comp.lean | sheet2.lean | sheet3.lean | sheet4.lean

src > 2020 > functions > injective_comp.lean

```lean
import tactic

open function

variables (X Y Z : Type) (f : X → Y) (g : Y → Z)

example : (injective f) ∧ (injective g) → injective (g ∘ f) :=
begin
  -- Assume that f and g are injective.
  rintro ⟨hf, hg⟩,
  -- We need to prove `g ∘ f` is injective.
  -- So say `a`, `b` are in `X`, and `(g ∘ f)(a) = (g ∘ f)(b)`
  rintro a b hab,
  -- We need to prove that `a = b`.
  -- By injectivity of `f`, it suffices to prove that `f(a) = f(b)`.
  apply hf,
  -- But this follows immediately from our assumption
  -- `g(f(a))=g(f(b))`,and injectivity.
  sorry
end
```

Lean Infoview

▼ injective_comp.lean:16:11

▼ Tactic state

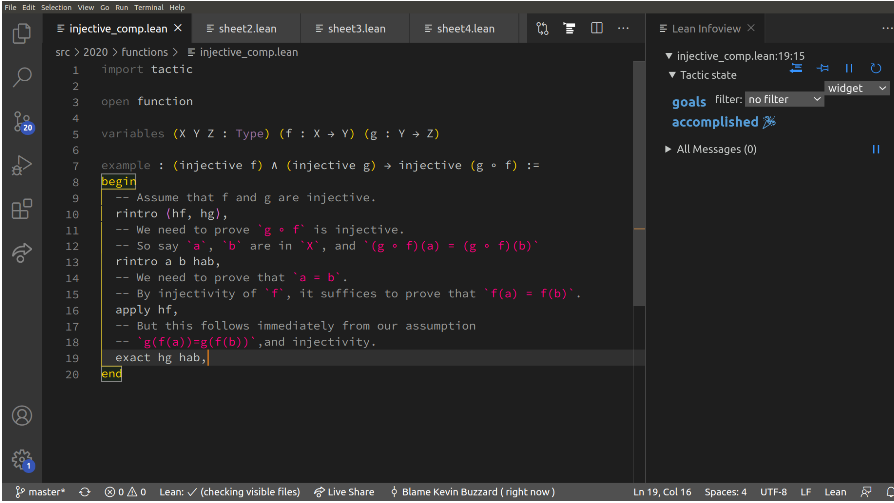
widget

1 goal filter: no filter

```
X Y Z : Type
f : X → Y
g : Y → Z
hf : injective f
hg : injective g
a b : X
hab : (g ∘ f) a = (g ∘ f) b
⊢ f a = f b
```

▶ All Messages (1)

master* ⓧ 0 ⚠ 1 Lean: ✓ (checking visible files) Live Share Blame Kevin Buzzard (3 days ago) Ln 16, Col 12 Spaces: 4 UTF-8 LF Lean

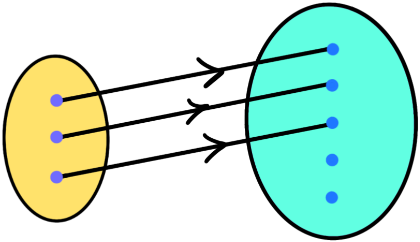

File Edit Selection View Go Run Terminal Help
injective_comp.lean
sheet2.lean
sheet3.lean
sheet4.lean
src > 2020 > functions > injective_comp.lean
import tactic
open function
variables (X Y Z : Type) (f : X → Y) (g : Y → Z)
example : (injective f) ∧ (injective g) → injective (g ∘ f) :=
begin
-- Assume that f and g are injective.
rintro ⟨hf, hg⟩,
-- We need to prove `g ∘ f` is injective.
-- So say `a`, `b` are in `X`, and `(g ∘ f)(a) = (g ∘ f)(b)`
rintro a b hab,
-- We need to prove that `a = b`.
-- By injectivity of `f`, it suffices to prove that `f(a) = f(b)`.
apply hf,
-- But this follows immediately from our assumption
-- `g(f(a))=g(f(b))`,and injectivity.
exact hg hab,
end
Lean Infoview
injective_comp.lean:19:15
Tactic state
widget
goals accomplished
filter: no filter
All Messages (0)
master*
0 0
Lean: ✓ (checking visible files)
Live Share
Blame Kevin Buzzard (right now)
Ln 19, Col 16
Spaces: 4
UTF-8
LF
Lean

Counting

Let's finish the video with some more counting.

Say X and Y are finite sets, and $f : X \to Y$ is an injective function.

Because f is injective, distinct elements of X get mapped to distinct elements of Y. So the size of Y must be at least the size of X.



Counting question 1) How many injections are there from a set of size 2 to a set of size 5?
Counting question 2) How many injections are there from a set of size 5 to a set of size 2?

Pause the video and have a go at these questions.

Counting question 1) How many *injections* are there from a set of size 2 to a set of size 5?

Remember that in the maths department, a function is determined by its values on all inputs.

Say the set of size 2 has element a and b. There are five places where we can send a. Now what?

Because we want f to be injective, we can't send b to the same place as a. So we only have four possibilities for $f(b)$.

So the total number of functions is $5 \times 4 = 20$.

Counting question 2) How many *injections* are there from a set of size 5 to a set of size 2?

Answer: none. For if there were an injection from a set of size 5 to a set of size 2, we would be able to deduce that $5 \leq 2$, which is false.

As an exercise, see if you can figure out the number of injections from a set of size m to a set of size p. Hint: there are p choices for the first element, $(p-1)$ for the second, $(p-2)$ for the third...