MAT224 – Linear Algebra II

Maps

Repeated Arguments

There are four topics in Linear Algebra essentially has the same logic but using different words with different understandings.

$$\begin{cases} \text{Linear Combination} \\ \text{Linear Equations} \\ \text{Subspaces} \\ \text{Linear Transformations} \end{cases}$$

Repeated Arguments

For example, being **linealy independent** , has different explainations in different perspective.

Logic	**uniqueness**
Linear Combination	**linealy independent**
Linear Equations	**no free variables**
Subspace	**is a direct sum**
Numerical	**no more than**
Linear Transformations	**injective**

They are essentially the same. Theorems with those words always has translation.

Repeated Arguments

The **uniqueness** of an described object means they are **no more than** 1.

The way of writting a vector into linear combinations of **linealy independent** vectors is **no more than** 1.

A linear equation with **no free variables** has **no more than** one solution.

If $W + U$ **is a direct sum** , the way of decomposing a vector $\vec{v} \in W + U$ as $\vec{v} = \vec{w} + \vec{u}$ with $\vec{w} \in W$ and $\vec{u} \in U$ is **no more than** 1.

A linear transformation is **injective** if the preimage of every element has **no more than** 1 element.

They are essentially the same, in slides we categorize them by symbol $\boxed{I}$.

Repeated Arguments

The interesting thing is each of the argument has its dual argument.

Logic	**existence**
Linear Combination	**span the whole space**
Linear Equations	**independent equations**
Subspace	**sum to the whole space**
Numerical	**no less than**
Linear Transformations	**surjective**

They are essentially the same and dual to previous one. Replacing all argument by dual argument will create another valid arguments.

Repeated Arguments

The **existence** of an described object means they are **no less than** 1.

The way of writting a vector into linear combinations of **span the whole space** vectors is **no less than** 1.

A linear equation with **independent equations** has **no less than** one solution.

If $W + U$ **sum to the whole space** , the way of decomposing a vector $\vec{v} \in W + U$ as $\vec{v} = \vec{w} + \vec{u}$ with $\vec{w} \in W$ and $\vec{u} \in U$ is **no less than** 1.

A linear transformation is **surjective** if the preimage of every element has **no less than** 1 element.

They are essentially the same, in slides we categorize them by symbol $\boxed{S}$.

Repeated Arguments

This slides categories most arguments into 7×2 categories

$$\text{related to } \textbf{uniqueness} \begin{cases} \boxed{I}\,\boxed{C_I}\,\boxed{D_I}\,\boxed{R_I} \\ \boxed{L_I}\,\boxed{N_I} \\ \boxed{I \mapsto S} \end{cases}$$

with its dual:

$$\text{related to } \textbf{existence} \begin{cases} \boxed{S}\,\boxed{C_S}\,\boxed{D_S}\,\boxed{R_S} \\ \boxed{L_S}\,\boxed{N_S} \\ \boxed{S \mapsto I} \end{cases}$$

Before rank-nullity theorem, you are essentially play with those 12 arguments.

Letter Meaning: C:composition, D: decomposition, L:Language, N:numerical, I:**injective** , S:**surjective**

Goal

In this slides, we will demonstrate those 14 categories of arguments in set theory. With this, all aguments in linear algebra is easy to remember. Arguments in the same categories has philosophically the same proof.

Shinchan likes the book

Shinchan likes reading books. 1:00pm - 9:59pm everyday is his reading time. He will read books 📚 hourly. Each day in the morning, he put a tag n to the book 📕 for reading during n:00pm-n:59pm.

n $\mapsto$ 📕

He likes a clean room, so he always put books in drawers.

📕, 📘, $\cdots \mapsto$ 

Maps

Those are described as maps mathematically.

Definition 1

Let X, Y be two sets, a **map** $f : X \longrightarrow Y$ is an assignment such that

1. (**existence**)For all $x \in X$, **there exists** $y \in Y$ associates to it. Denoted by $y = f(x)$
2. (**uniqueness**)For all $x \in X$, the element $y = f(x) \in Y$ associates to it is **unique** .

Definition 2

The set X in a map $f : X \longrightarrow Y$ is called **domain**, Y is called **codomain**

Maps

The way Shinchan puts tags to books he read is a **map**

$$f : \{1, \cdots, 9\} \longrightarrow \{\text{📕}, \text{📘}, \cdots\}$$

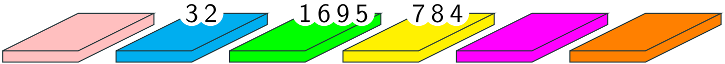
$$n \longmapsto \text{The book he read during n:00pm-n:59pm}$$

Because in our assumption, he reads book hourly, so he will read the same book during each hour(**uniqueness**).

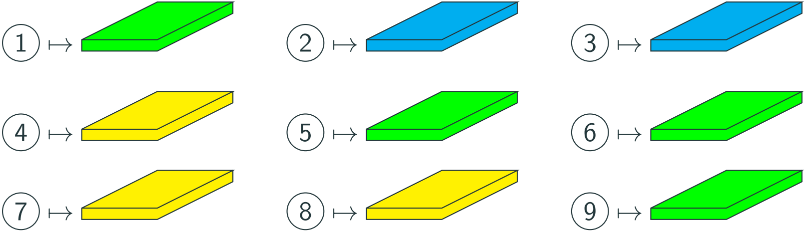
On the other hand, he likes reading books, each hour during 1:00-9:59 he will read some book, so the tags in domain always associate to a book(**existence**).

Maps

This is an example of how exactly this map looks like(way 1)

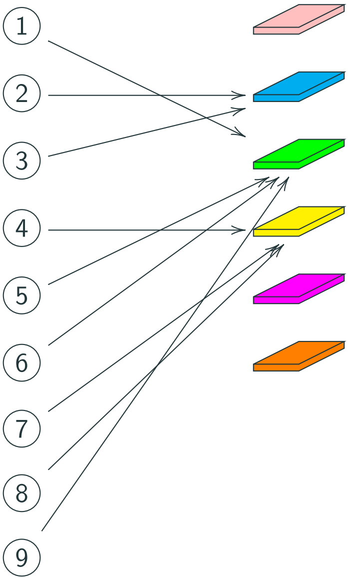


This means (way 2)



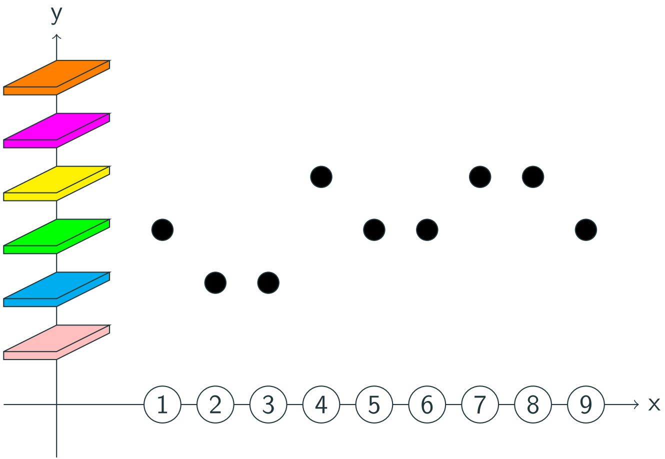
Maps

Some people also like to think a map like (way 3)



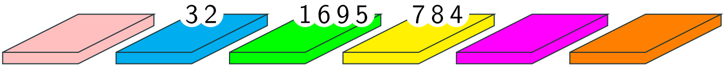
Maps

Some people think map as a graph of a function(way 4)



Maps

NO! JUST THINK ALL MAPS in Linear Algebra LIKE WAY 1.



Other ways of thinking will **CONFUSE** you.

Maps

The way Shinchan put books to drawers is a **map**

$g : \{$ , , $\cdots\} \longrightarrow$ 

 $\longmapsto$ The drawer he want to put this book in

Because he can not put one book into two drawers, he loves book, do not want to tear it into two part (**uniqueness**).

Everybook has to be in a drawer after reading, because he want his room clean.(**existence**).

Maps

This is an example of how exactly one of such a map looks like

Descriptive and Constructive

We start with $\boxed{L_I}$, $\boxed{L_S}$

In math, Elements in **codomain** are used in **descriptive language** . Elements in **domain** are used in **constructive language** . Let's start with examples to see what it mean.

Descriptive Language and codomain

Shinchan's mom, Misae, find a book 📘somewhere. She says

Misae:*Hey, what is this book* 📘*?*

Shinchan:*This is* **a** *book* **in my comics drawer***.*

In this situation, Shinchan is describing this book. We call this way of describing an objects as **descriptive language** $\boxed{L_l}$.

Descriptive Language and codomain

descriptive language classifies an object to a **class** . For example.

She is **a** *girl with* **long,beautiful, and black hairs**.

Class: Girls with long, beautiful, black hairs.

Today is **a sunny** *day.*

Class: dates that is sunny.

In English, objects described by **descriptive language** always uses the article **a** or **an**.

Descriptive Language and codomain

In the book to drawers map, using each drawer to describe a book is an **descriptive language** .

$g \quad : \quad \{$, $, \cdots\} \quad \longrightarrow$ 

 $\quad \longmapsto \quad$ The drawer he want to put this book in

Example:**a** *book x in* **comics drawer**

Mathematical:$x \in \{x : g(x) \in$ **comics drawer**$\}$

Example:**a** *book x in* **novels drawer**

Mathematical:$x \in \{x : g(x) \in$ **novels drawer**$\}$

In math, **using** elements or subsets in **codomain** to describe, classifies, or give properties of elements in domain is a **descriptive language** .

Descriptive Language and codomain

On the other hand, any collections of objects by **descriptive language** can be represented by **preimage** of some subset of a map.

Definition 3

$\boxed{L_I}$ For any map $f : X \longrightarrow Y$ and any subset $S \subset Y$, the **preimage** of S is defined by

$$f^{-1}(S) := \{x \in X : f(x) \in S\}$$

Descriptive Language and codomain

Another mathematical **descriptive language** is like the following:

She is **a** *girl with* **long,beautiful, and black hairs**.

$$S = \{\text{long hair}, \text{beautiful hair}, \text{blak hair}\}$$

$$\text{She} \in \{x \in \text{Set of girls} : x \text{ has property } p \textbf{ for all } p \in S\}$$

If you see **for all** appears, it is likely to be an **descriptive language** .

Descriptive Language and codomain

Any person with **black or gold hairs** is an elemnt of the preimage of {black,gold} of the following classification maps.

$$\begin{aligned} f \quad : \quad \text{Set of people} \quad &\longrightarrow \quad \text{Set of colors} \\ x \quad &\longmapsto \quad \text{hair color of } x \end{aligned}$$

Any objects described by **descriptive language** can be written as a **preimage** of **the subset of desired properties** under a map

$$\begin{aligned} f \quad : \quad \text{Set of described objects} \quad &\longrightarrow \quad \text{Classification Set} \\ x \quad &\longmapsto \quad \text{Properties of } x \end{aligned}$$

Descriptive Language and codomain

Insufficient **descriptive language** may describe more than one elements, the described object may not be **unique** .

A student in university of toronto

More **descriptive** words let the object more close to **uniqueness** .

A boy in university of toronto studying MAT224

A perfect **descriptive language** describe a unique object(exactly one).

A boy in university of toronto studying MAT224 now and smiling

Too many **descriptive language** may contradicts each other so no element saitiesfies our description at all.

A boy in university of toronto studying MAT224 now and smiling in the first row

Descriptive Language and codomain

$$
\begin{tikzcd}
\text{Descriptive words:} & \text{insufficient} \arrow[r, dashed, "\text{more}", "\text{words}"'] \arrow[drr] & \text{sufficient} \arrow[r, dashed, "\text{more}", "\text{words}"'] \arrow[d] & \text{contradictive} \arrow[dll] \\
\text{Described Objects:} & \text{No object} \arrow[r, dashed, "\text{more}", "\text{objects}"'] & \text{Exactly one} \arrow[r, dashed, "\text{more}", "\text{objects}"'] & \text{More than one}
\end{tikzcd}
$$

Constructive Language and Domain

Shinchan calls his mom to pick up some book for him.

Shinchan:*Mom, give me* **the** *book* **to be read at 3:00-3:59***.*

Misae:*Here you are*  *!*

In this situation, Shinchan is specifying this book. We call this way of specifying an objects as **constructive language** $\boxed{L_S}$.

Constructive Language and Domain

constructive language points out or construct an object directly by **parameters** . For example

She is **John's** *mother*
She is **the** *mother of* **John**

The *owner* **of the apartment CASA**.

Here John, CASA are **parameters** to describe a person.

In English, objects described by **constructive language** are always using article **the**.

Constructive Language and Domain

In the tags to books map, using each tag to describe a book for specific reading time is an **constructive language** .

$$f \quad : \quad \{⓵, \cdots, ⓽\} \quad \longrightarrow \quad \{$$  ,  $$, \cdots \}$$

$$ⓝ \quad \longmapsto \quad \text{The book he read during n:00pm-n:59pm}$$

Example:**The** ***book*** **of 3:00pm-3:59pm reading time**$= f\left(③\right)$

Example:**The** ***book*** **of 5:00pm-5:59pm reading time**$= f\left(⑤\right)$

In math, **using** elements or subsets in **domain** to specify, parametrize, or constructs elements directly in codomain is a **constructive language** $\boxed{L_S}$.

Constructive Language and Domain

On the other hand, any collections of objects by **constructive language** can be represented by **image** of some subset of a map.

Definition 4

$\boxed{L_S}$ For any map $f : X \longrightarrow Y$ and any subset $S \subset X$, the **image** of S is

$$f(S) := \{y \in Y : y = f(x) \text{ for some } x \in S\}$$

Note: Onece the word *for some* appears, it is likely a **constructive language** .

Constructive Language and Domain

The books of Shinchan to read from 2:00pm-5:59pm is described by the image of $\{2, 3, 4, 5\}$ under the following map

$$
\begin{array}{rcccl}
f & : & \{1, \cdots, 9\} & \longrightarrow & \{\text{📕}, \text{📘}, \cdots\} \\
 & & n & \longmapsto & \text{The book he read during n:00pm-n:59pm}
\end{array}
$$

In the same way, any objects described by **constructive language** can be written as a **image** of a map. By

$$
\begin{array}{rcccl}
f & : & \text{Set of parameters} & \longrightarrow & \text{Set of described objects} \\
 & & \text{parameter} & \longmapsto & x
\end{array}
$$

Constructive Language and Domain

Insufficient **constructive language** may not enough to specify all objects, the parameters for some objects may not **exists** .

Shinchan's mom

More **constructive** words let each object more close to get **existence** of parameters.

Shinchan's mom, Makuro's mom, Nobita's mom, Jack's mom,$\cdots$

The **constructive language** pursue **existence** of **parameters** . A perfect **constructive language** gives every object a **parameters** .

$\cdots$'s mom, $\cdots$'s mom, $\cdots$'s mom,$\cdots$(All mom on earth listed)

Too many **constructive language** may redundantly use multiple parameters to specify the same object.

A's mom, B's mom, $\cdots$'s mom,$\cdots$(A and B are brothers)

Constructive Language and Domain

The following diagram use the short *pa* for **parameters** . Use *suff* for sufficient, use *eco* for economical.

$$
\begin{tikzcd}
\text{Parametrization:} & \text{insuff} \arrow[r, dashed, "\text{more}", "\text{parameter}"'] \arrow[d] & \text{suff and eco} \arrow[r, dashed, "\text{more}", "\text{parameter}"'] \arrow[d] & \text{redundant} \arrow[d] \\
\text{An Object:} & \text{may have 0 pa} \arrow[r, dashed, "\text{more}", "\text{parameter}"'] & \text{Exactly 1 pa} \arrow[r, dashed, "\text{more}", "\text{parameter}"'] & \text{may more than 1 pa}
\end{tikzcd}
$$

Injective and Surjective

Now we start talking $\boxed{I}$, $\boxed{S}$. Note: We say object **unique** means it is **at most one** , means not exists is also considered as **unique** .

Injective and Surjective

In the viewpoint of **classification**:

injective is **uniqueness** of described object for **each** class.

Definition 5

$\boxed{I}$ A map $f : X \longrightarrow Y$ is **injective** if the **preimage** of each element in Y has **at most one** element.

This means objects in each class is **unique** (but may not **exists**).

Injective and Surjective

In the viewpoint of **classification**: **surjective** is **existence** of described object for **each** class.

Definition 6

$\boxed{S}$ A map $f : X \longrightarrow Y$ is **surjective** if the **preimage** of each element in Y has **at least one** element.

This means objects in each class **exists** (but may not **unique**)

Injective and Surjective

In the viewpoint of **parametrization** :

surjective is **sufficiency** of **parameters** .

Definition 7

$\boxed{S}$ A map $f : X \longrightarrow Y$ is **surjective** if for every $y \in Y$, there **exists** a parameter $x \in X$ with $y = f(x)$

This means Obejcts in Y are sufficiently parametrized by X: every element in Y has parameter.

Injective and Surjective

In the viewpoint of **parametrization** :

injective and is **ecnomical** of **parameters** .

Definition 8

$\boxed{I}$ A map $f : X \longrightarrow Y$ is **injective** if for every $x_1 \neq x_2$, we have $f(x_1) \neq f(x_2)$

This means Obejcts in Y are **ecnomical** parametrized by X: no parameters wasted.

Injective and Surjective

There is a way of describing **surjective** using **constructive language** .

Definition 9

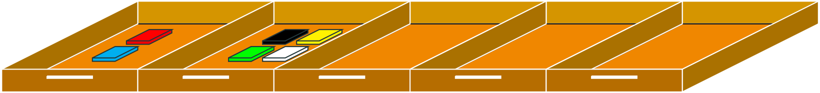
The image of a map $f : X \longrightarrow Y$ is $f(S)$ the image of the whole domain. In other words,

$$\mathrm{Im}(f) = \{y \in Y : y = f(x) \text{ for some } x \in X\}.$$

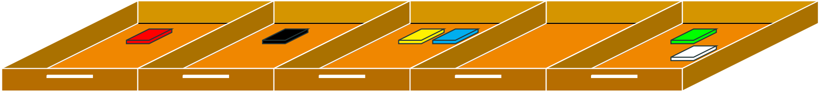
In viewpoint of **parametrization** , elements in X are **parameters** of Y. $\mathrm{Im}(f)$ are the **sub**set of all parametrized element in Y.

Injective and Surjective

Image is measureing **how close** is the map to be a **surjective** . The more element the image have, the more the map close to a **surjective**



Well, your Image is relatively small ↑



Good job! Image gets bigger! more close! ↑

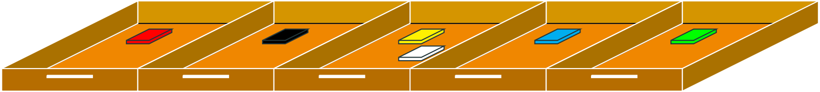


Image is biggest! Congradulations!

You have been awarded the **meadal** *of* **surjective** *!*

Proposition 1

$\boxed{L_S}\boxed{S}$ A map $f : X \longrightarrow Y$ is **surjective** if and only if

$$\mathrm{Im}(f) = Y$$

That is, a map is **surjective** if the subset of parametrized elements reached biggest possible (so it fills the whole Y)

Injective and Surjective

There is a way of describing **injective** using **descriptive language** .

Definition 10

A **quotient**set of a set X is a classification of elements in X, that is a subset of the $\mathcal{P}(X)$**:set of subset of X**

$$Q \subset \mathcal{P}(X)$$

such that any two elements in Q do not have intersection, the union of elements in Q equals to X.

Example:

$\{\{1,2,5\},\{3,4\}\}$ is a quotientset of $\{1,2,3,4,5\}$ it classifies $1,2,5$ to a class, and $3,4$ to a class.

Definition 11

The **coImage** $\mathrm{coIm}(f)$ of a map $f : X \longrightarrow Y$ is a quotientset of X classified by elements in YIn other words, the set of preimages of elements in Y

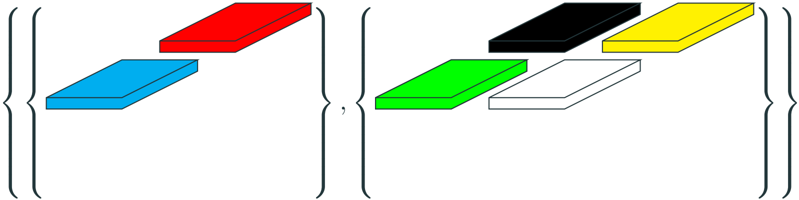
$$\mathrm{coIm}(f) = \{f^{-1}(\{y\}) : \text{ for some } y \in Y\}$$

Injective and Surjective

Too abstract! Let's talk with example!

In this map

The coimage is



Coimage means classification.

Injective and Surjective

Subset we can compare who is bigger who is smaller, for example, subsets of $\{1,2,3,4,5\}$

$\{1,2\}$ smaller subset $\qquad$ $\{1,2,3\}$ bigger subset

In quotientset we comprare who is more precise who is less precise

$\{1,2,3,4,5\}$ not precise at all $\qquad$ $\{\{1,2,3\},\{4,5\}\}$ more precise

$\{\{1\},\{2\},\{3\},\{4\},\{5\}\}$ most precise

Injective and Surjective

Proposition 2

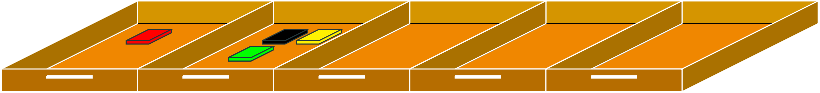
$\boxed{L_I}$ $\boxed{I}$ A map $f : X \longrightarrow Y$ is **injective** if and only if

$$\mathrm{coIm}(f) = X$$

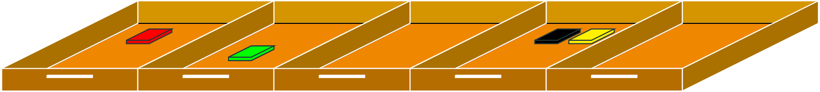
The above proposition, means a map is **injective** if the classification of elements in X reached most preciseded possible (so every class has a single element)

Injective and Surjective

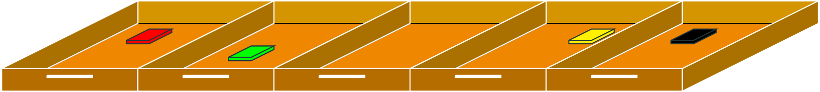
coImage is measureing **how close** is the map to be a **injective** . The more precise the coImage have, the more the map close to a **injective**



Well, coImage relatively not so precise ↑



Good job! coImage gets more preciser! more close! ↑



coImage is the precisest! Congradulations!
You have been awarded the **meadal** *of* **injective** *!*

Injective and Surjective

In Linear Algebra, *coimage is domain* $\Leftrightarrow$ *kernel is zero.*

The relation of coImage and kernel is related by **quotient**space.

Composition of maps

Now we start talking $\boxed{C_I}$, $\boxed{D_I}$, $\boxed{C_S}$, $\boxed{D_S}$.

Composition of maps

Use the previous example before. Remember each book can be **no tag**(For example: Shinchan do not want to read this book today), or tagged **more than onece** (For example: If Shinchan wants to read the book for the whole day, he put all tags on it).

Consider

$$X = \text{Set of number tags} = \{1, 2, 3, \cdots\}$$

$Y = \text{Set of books} = \{$ , , $\cdots\}$

$Z = \text{Set of drawers} =$ 

Composition of maps

Remember the map $f : X \longrightarrow Y$ associate tags to books

$$①, ② \mapsto$$ 

.

Elements in X are **parameters** of books in Y.

Remember the map $g : Y \longrightarrow Z$ associate books to drawers. Elements in Y are **class** of books in Y.

 $, \cdots \mapsto$ 

Composition of maps

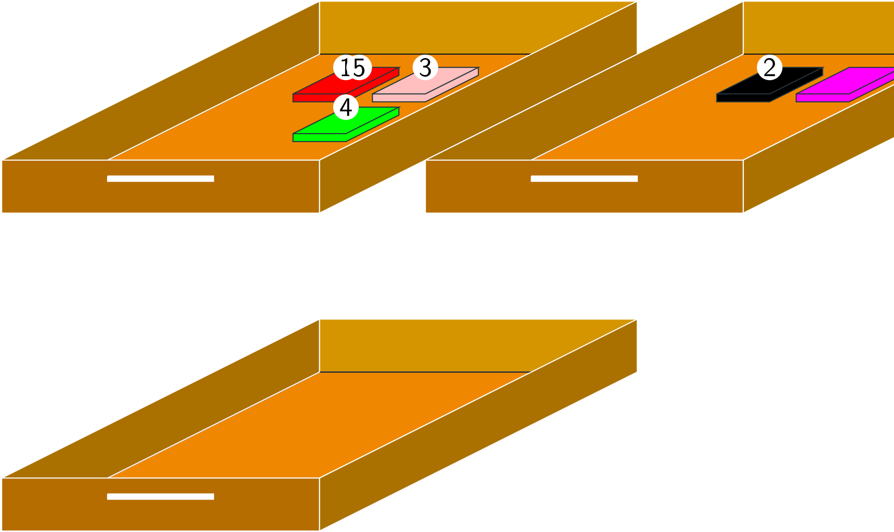
As we put all books into drawers, the tags has also been put into drawers **via books**. The map of tags put into drawers is described by composition

$$g \circ f : X \longrightarrow Z$$

$$g \circ f : \{1, 2, \cdots\} \longrightarrow$$

Composition of maps

This is an intuitive picture of composing two maps



Three important properties of injective

We are ready to describe $\boxed{I}\boxed{C_I}\boxed{D_I}$

Three important properties of injective

Let's firstly translate the meaning of **injective** in this senario.

The map $f : X \longrightarrow Y$

$f : \{1, 2, \cdots\} \longrightarrow \{$📚, 📚, $\cdots\}$

is **injective** means each book only has one tag or no tag.

The map $g : Y \longrightarrow Z$

$g : \{$📚, 📚, $\cdots\} \longrightarrow$ 

is **injective** means each drawer only has one book or no book.

Three important properties of injective

The map $g \circ f : X \longrightarrow Z$

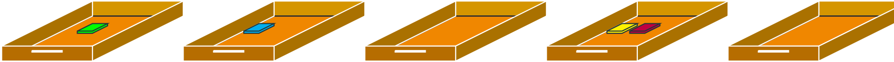
$$g \circ f : \{①, ②, \cdots\} \longrightarrow$$ 

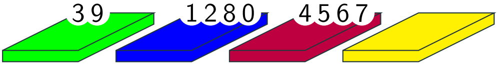
is **injective** means each drawer only has one tag or no tag.

Three important properties of injective

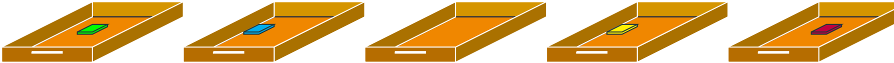
Three important properties of injective

Which of the following maps is an injective, why? Map 1:



Map 2: 

Map 3:



Map 4: 

Three important properties of injective

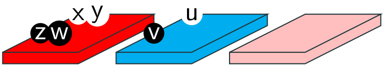
Proposition 3

$\boxed{I}$ **left cancellation** holds for **injective** maps. That is, if $g : Y \longrightarrow Z$ is **injective** , then for any $f_1 : X \longrightarrow Y$ and $f_2 : X \longrightarrow Y$ such that $g \circ f_1 = g \circ f_2$, then $f_1 = f_2$.

for simplicity, assume Shinchan today only read from 1:00pm-3:59pm

Proof Step 1

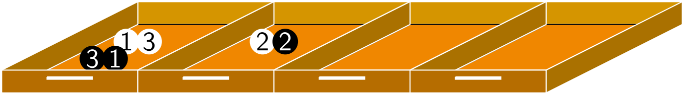
Shichan use white tags (g_1) {1, 2, 3} for today's reading. Black tags(g_2) {1, 2, 3} for tomorrows reading.



Three important properties of injective

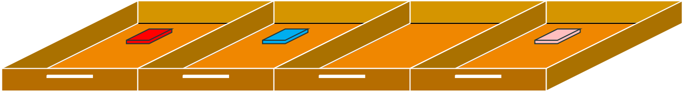
Proof Step 2

For any drawer, onece it has white (n) the black tag (n) must be in the same drawer, vise versa.($g \circ f_1 = g \circ f_2$)



Three important properties of injective

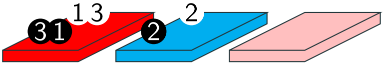
But each drawer only has at most one book. (g is an **injective**)



Three important properties of injective

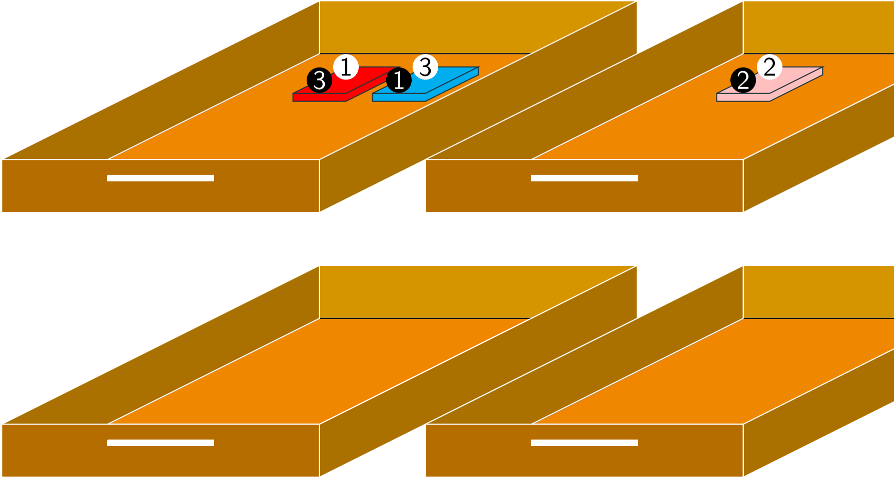
Proof Step 3

Then it must be that every black tags n and white tags n with the same number must attached to the same book.($g_1 = g_2$)



Three important properties of injective

counter example: if g is not an **injective** , $g \circ f_1 = g \circ f_2$ may not implies $f_1 = f_2$



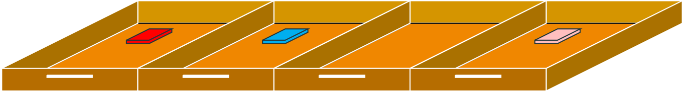
Three important properties of injective

Proposition 4

$\boxed{C_I}$ The composition of **injective** maps is an **injective** .

(Today Shinchan's reading hour is 1:00-2:59) **Proof Step 1**

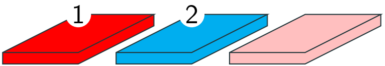
If each drawer only has one book or no book,



Three important properties of injective

Proof Step 2

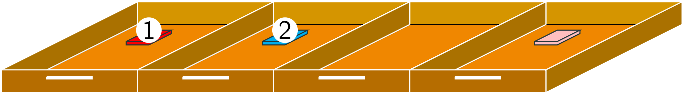
each book only has one tag or no tag,



Three important properties of injective

Proof Step 3

then each drawer only has one tag or no tag on it.



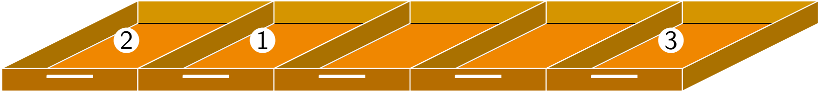
Three important properties of injective

Proposition 5

$\boxed{D_I}$ If the composition $g \circ f : X \longrightarrow Z$ is an **injective** , then the **right factor** f is an **injective**

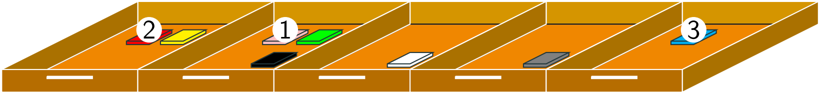
Proof Step 1

Each drawer only has one tag or no tag.($g \circ f : X \longrightarrow Z$ is **injective**)



Proof Step 2

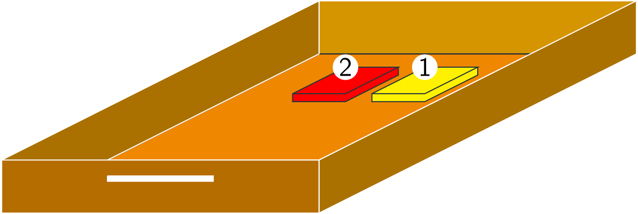
So each book can only has one tag or no tag.(f is an **injective**)



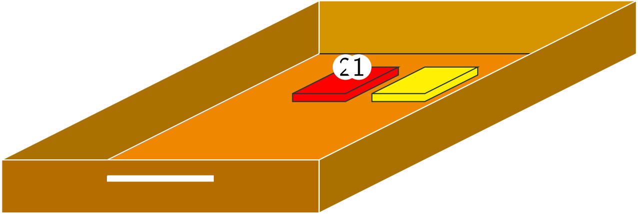
(In this specific example, you can also see the **left factor** g may **not** be an **injective** .)

Three important properties of injective

Counter example if $g \circ f$ is not an **injective** , then f **may** be an **injective** ,



or maybe not.



Three important properties of surjective

We are ready to describe $\boxed{S}\boxed{C_S}\boxed{D_S}$.

Three important properties of surjective

Let's firstly translate the meaning of **surjective** in this senario.

The map $f : X \longrightarrow Y$

$$f : \{①, ②, \cdots\} \longrightarrow \{📚, 📚, \cdots\}$$

is **surjective** means all books are tagged. No untagged book.

The map $g : Y \longrightarrow Z$

$$g : \{📚, 📚, \cdots\} \longrightarrow$$ 

is **surjective** means all drawers has books. No empty drawers.

Three important properties of surjective

The map $g \circ f : X \longrightarrow Z$

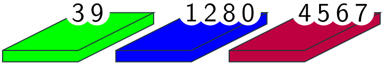
$$g \circ f : \{1, 2, \cdots\} \longrightarrow$$

is **surjective** means all drawers has tags.

Three important properties of surjective

Which of the following maps are a **surjective** , why?

Map 1: 

Map 2: 

Map 3:

Map 4: 

Three important properties of surjective

Proposition 6

$\boxed{S}$ **right cancellation** holds for **surjective** maps. That is, if $f : X \longrightarrow Y$ is **surjective** , then for any $g_1 : Y \longrightarrow Z$ and $g_2 : Y \longrightarrow Z$ such that $g_1 \circ f = g_2 \circ f$, then $g_1 = g_2$.

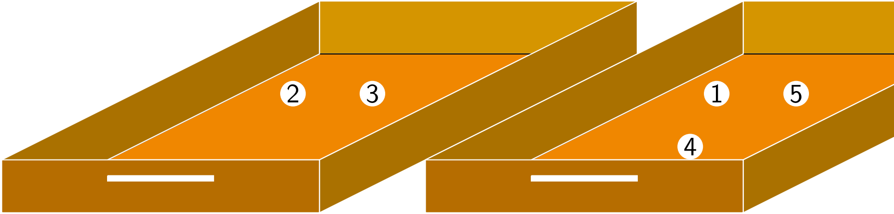
Proof step 1

Shichan puts books randomly in drawers everyday. He compared how he put books into drawers today(g_1) and yesterday(g_2).

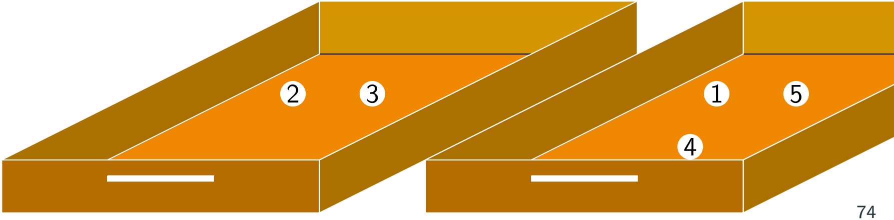
Three important properties of surjective

Proof step 2: Since every tag goes to the same drawer today and yesterday.($g_1 \circ f = g_2 \circ f$)

Today:

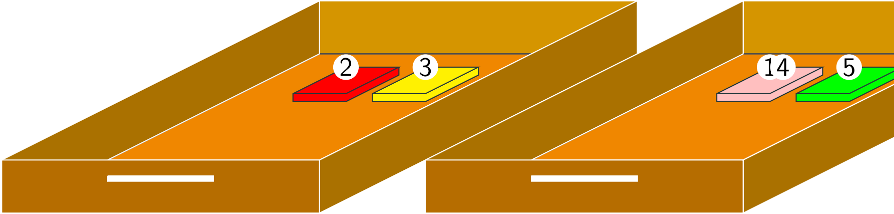


Yesterday:

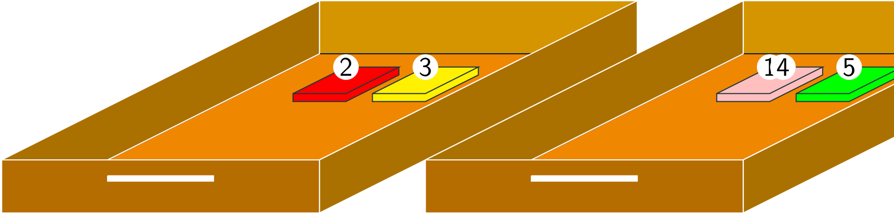


Three important properties of surjective

Proof step 3: Since all books are tagged(f **surjective**), this means each book has to go to the same drawer today and yesterday.($g_1 = g_2$) Today:

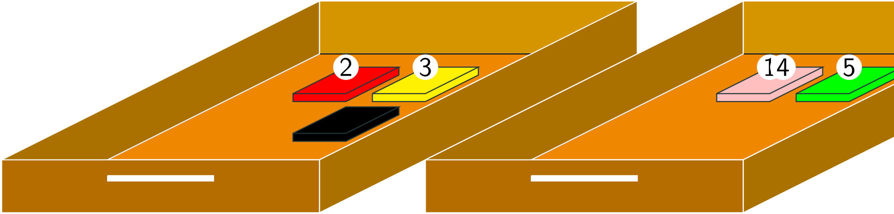


Yesterday:

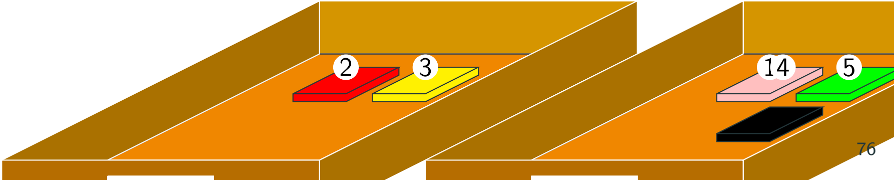


Counter Example: **right cancellation** may be false if f is not a **surjective** , this means it could be $g_1 \circ f = g_2 \circ f$, but $g_1 \neq g_2$ (The untagged book is free to go anywhere)

Today:



Yesterday:



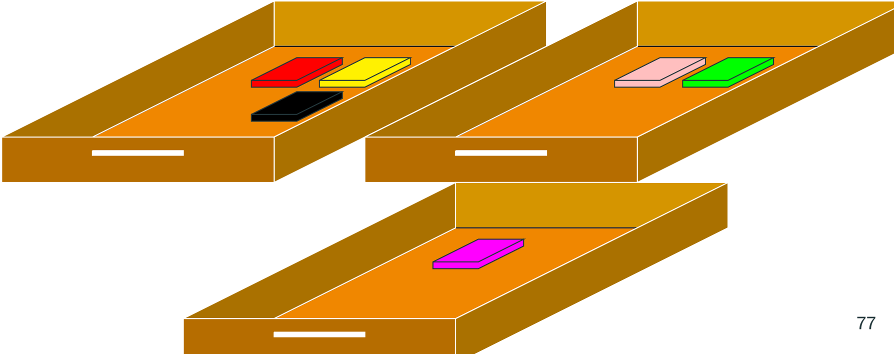
Three important properties of surjective

Proposition 7

$\boxed{C_S}$ The composition of **surjective** maps is an **surjective** .

Proof step 1

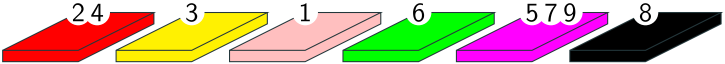
All drawers has books,



Three important properties of surjective

Proof step 2

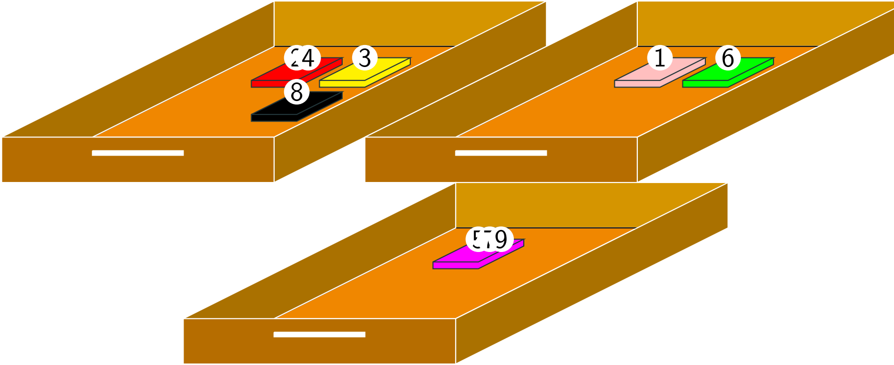
All books has tags,



Three important properties of surjective

Proof step 2

So all drawers has tags.



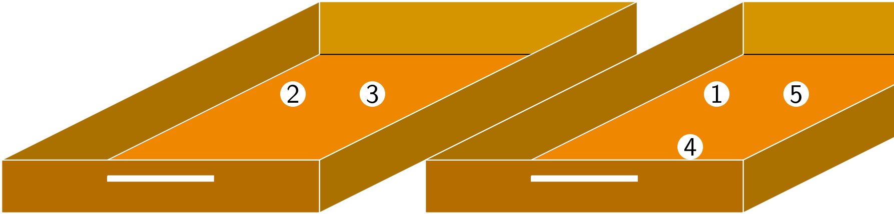
Three important properties of surjective

Proposition 8

$\boxed{D_S}$ If the composition $g \circ f : X \longrightarrow Z$ is an **surjective** , then the **left factor** g is an **surjective**

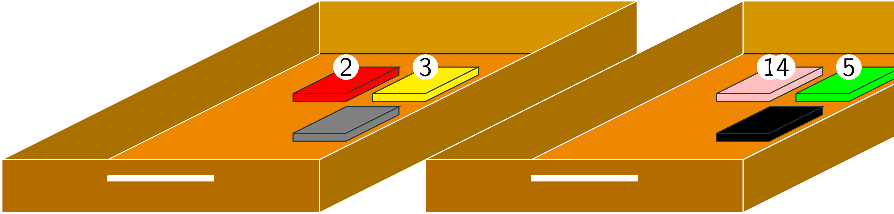
Proof step 1

All drawers has tags.($g \circ f : X \longrightarrow Z$ is **surjective**)



Proof step 2

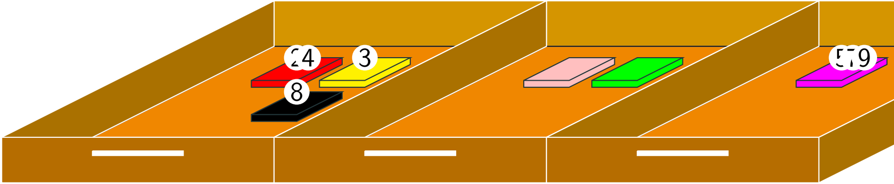
So all drawers has books.(g is an **surjective**)



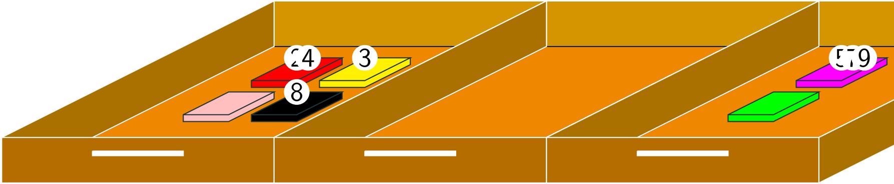
(In this example, you may discover untagged books, so the **right factor** f may not be a **surjective** .)

Three important properties of surjective

A counter example if $g \circ f$ is not **surjective** , then g may be a **surjective** :



or g may not be a **surjective**



Three important properties of surjective

Next we describe $\boxed{N_I}$, $\boxed{N_S}$. Note this only happens for finite set!!!(It would fail without finiteness)

Three important properties of surjective

Proposition 9

$\boxed{N_I}$ If there is an **injective** $f : X \longrightarrow Y$ between finite sets X, Y, then number of element in X is **no more than** number of element in Y.

If every drawer has **at most one** book,

Then books is **no more than** than drawers.

Three important properties of surjective

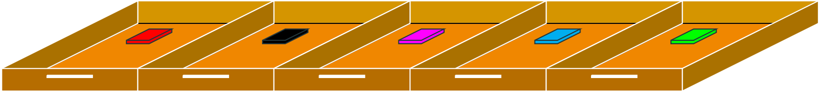
Proposition 10

$\boxed{N_I}$ If there is an **injective** $f : X \longrightarrow Y$ between finite sets X, Y of the same elements, then f is a **isomorphism** , in particular an **surjective**

Shinchan has drawers as many as books,

every drawer has **at most one** book,

so every drawer must have **exactly one** book.

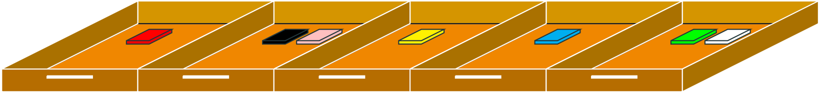


Proposition 11

$\boxed{N_S}$ If there is an **surjective** $f : X \longrightarrow Y$ between finite sets X, Y, then number of element in X is **no less than** number of element in Y.

If every drawer has **at least one** book,

Then books is **no less than** drawers.



Three important properties of surjective

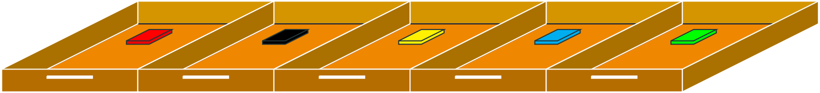
Proposition 12

$\boxed{N_S}$ If there is an **surjective** $f : X \longrightarrow Y$ between finite sets X, Y of the same elements, then f is a **isomorphism** , in particular an **injective**

Shinchan has drawers as many as books,

every drawer has **at least one** book,

so every drawer must have **exactly one** book.



Duality

We give examples for $\boxed{S \mapsto I}$ and $\boxed{I \mapsto S}$, those connections of **uniqueness** and **existence** is the most hardest one in logics of Linear Algebra.

Duality

Shinchan is used to put books in random drawers every day.

Misae do not like his way, she uses the same number of tags as Shichan $\{1, 2, \cdots, 9\}$, everyday Misae put those tags on drawers in a random way.

She asks Shinchan must put the book he read at n:00-n:59 to the drawer with her lable n.

Duality

Misae put tags randomly on drawer is a map

$h : \{1, 2, \cdots\} \longrightarrow$ 

Shinchan has his own tags

$f : \{1, 2, \cdots\} \longrightarrow \{$  ,  $, \cdots\}$

Duality

Shinchan wants to put books in drawers

$g : \{$📚, 📚, $\cdots\} \longrightarrow$ 

To make sure the tags agree, we must have $h = g \circ f$

Proposition 13

$\boxed{I \mapsto S}$ Suppose $f : X \longrightarrow Y$ is an **injective** , for any map $h : X \longrightarrow Z$, there always **exists** a map $g : Y \longrightarrow Z$, such that $h = g \circ f$.

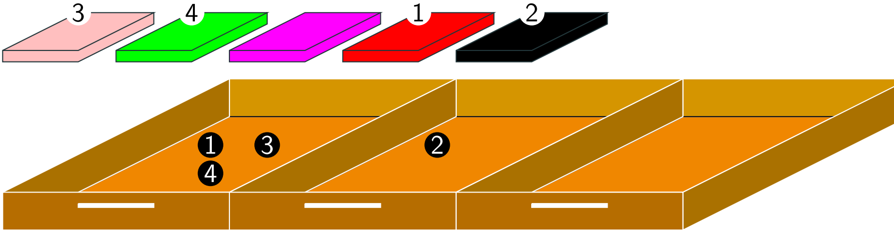
Proof step 1

Each book only has one tag or no tag.(**injective**)

Duality

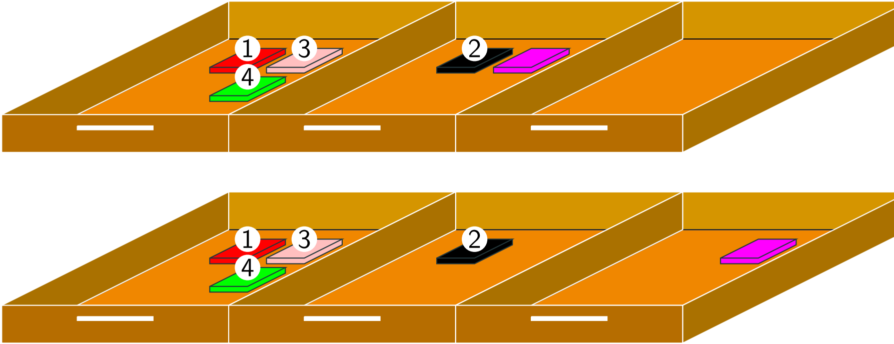
Proof step 2

So Shinchan only need to put those tagged book to the specific drawer prescribed by Misae. He can always make it.(**exists**)



Comments s

There are also even multiple ways to do it because untagged book can go to any drawer.(may not **unique**)

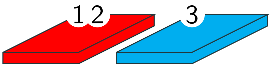


Duality

Proposition 14

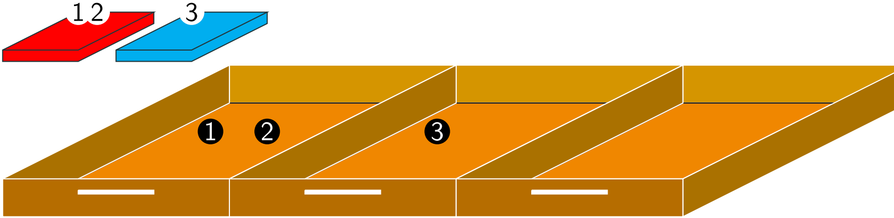
$\boxed{S \mapsto I}$ Suppose $f : X \longrightarrow Y$ is an **surjective** , for any map $h : X \longrightarrow Z$, a map $g : Y \longrightarrow Z$ such that $h = g \circ f$ must be **unique** .(means it also possible to not exist)

Proof step 1

All books are tagged.(**surjective**) 

Proof step 2

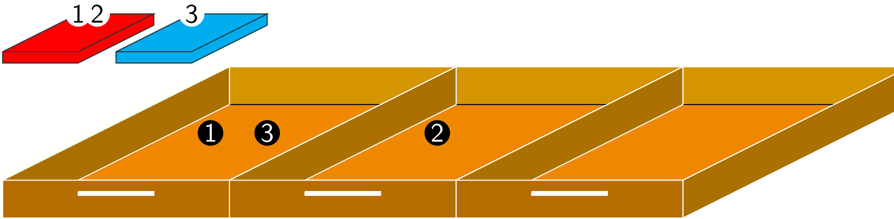
So all book has to put in a specific drawers prescribed by Misae. There is no second way to do it because there is no untagged book.(**unique**)



Duality

Comments

But there maybe a book with two tags, which Misae could asked it to put it into two different drawers, in this case Shinchan can not finish Misae's task.(may not **exists**)



Left and Right inverse

There is also some other properties $\boxed{R_I}$, $\boxed{R_S}$ regarding inverse.

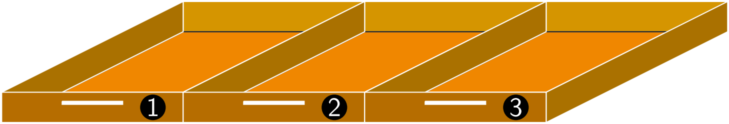
Left inverse

Proposition 15

$\boxed{R_l}$ $f : X \longrightarrow Y$ is an **injective** if and only if it has the **left inverse**, which means there exists $g : Y \longrightarrow X$ such that $g \circ f = \mathrm{id}$

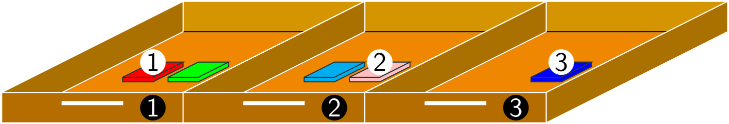
We use numbers, double them in pairs ❶,① ; ❷,② ; ❸,③

Put black tags on drawers, give white tags to Shinchan



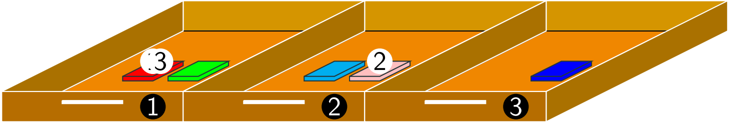
Left inverse

A map $f : X \longrightarrow Y$ hs a **left inverse** means each tagged book can go to the same tagged drawer.



Left inverse

If f is not an **injective** , then f can not have **left inverse** , since there are some book with two different numbers on it.

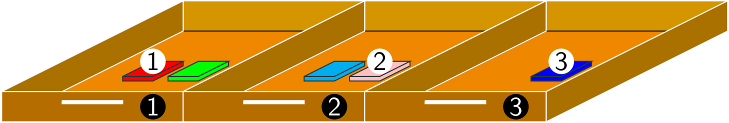


Right Inverse

Proposition 16

$\boxed{R_I}$ $g : Y \longrightarrow Z$ is an **surjective** if and only if it has a **right inverse** , which means there exists $f : Z \longrightarrow X$ such that $g \circ f = \mathrm{id}$

A map $g : Y \longrightarrow Z$ has a **right inverse** means we can **select a book** to have the same tag with drawer.



Clear this **selection** exists if and only if g is a **surjective** .

Right Inverse

If g is not **surjective** , then we can not select books for the empty drawer.