Welcome to CS103!

- ***Handouts!***
  - Course Syllabus
- ***Today:***
  - Course Overview
  - Introduction to Set Theory
  - The Limits of Computation

Are there “laws of physics” in computer science?

Introduction to Set Theory

Key Questions in CS103

- What problems can you solve with a computer?
  - ***Computability Theory***
- Why are some problems harder to solve than others?
  - ***Complexity Theory***
- How can we be certain in our answers to these questions?
  - ***Discrete Mathematics***

Instructor

Amy Liu (***liuamyj@cs.stanford.edu***)

TAs

Hugo Valdivia (***hugov65@stanford.edu***)
Amanda Spyropoulos
Gili Rusak
Teresa Noyola

Staff Email List: ***cs103-sum1819-staff@lists.stanford.edu***

Course Website

https://cs103.stanford.edu

Prerequisite / Corequisite

CS106B

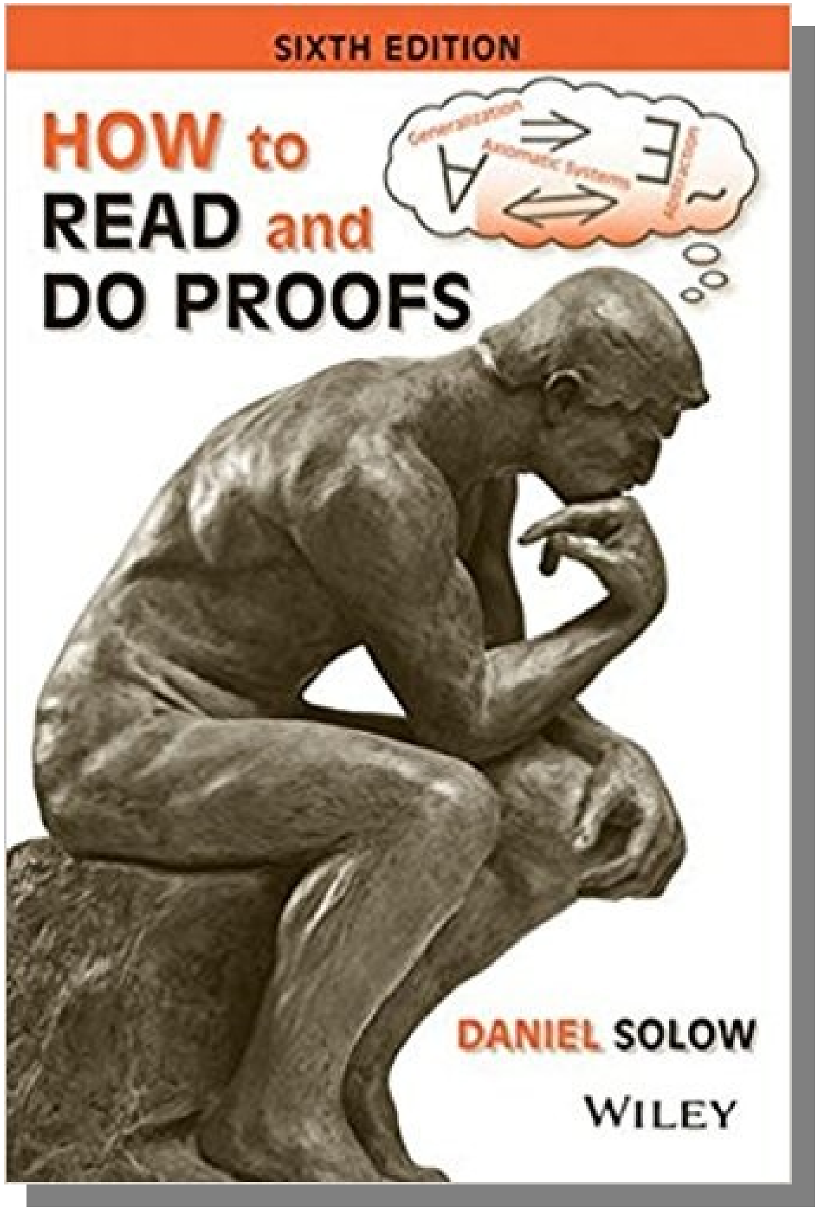
The problem sets throughout the quarter will have some programming assignments. We'll also reference some concepts from CS106B/X, particularly recursion, throughout the quarter.

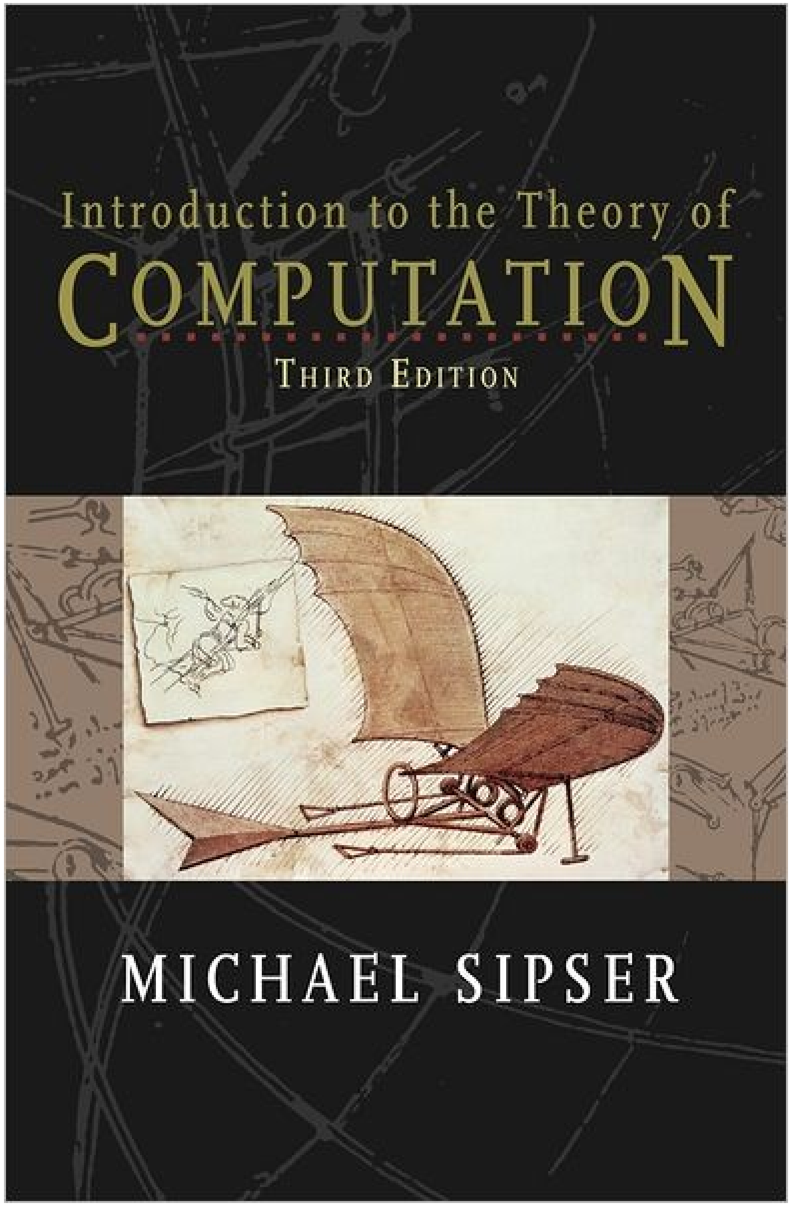
There aren't any math prerequisites for this course – high-school algebra should be enough!

Problem Set 0

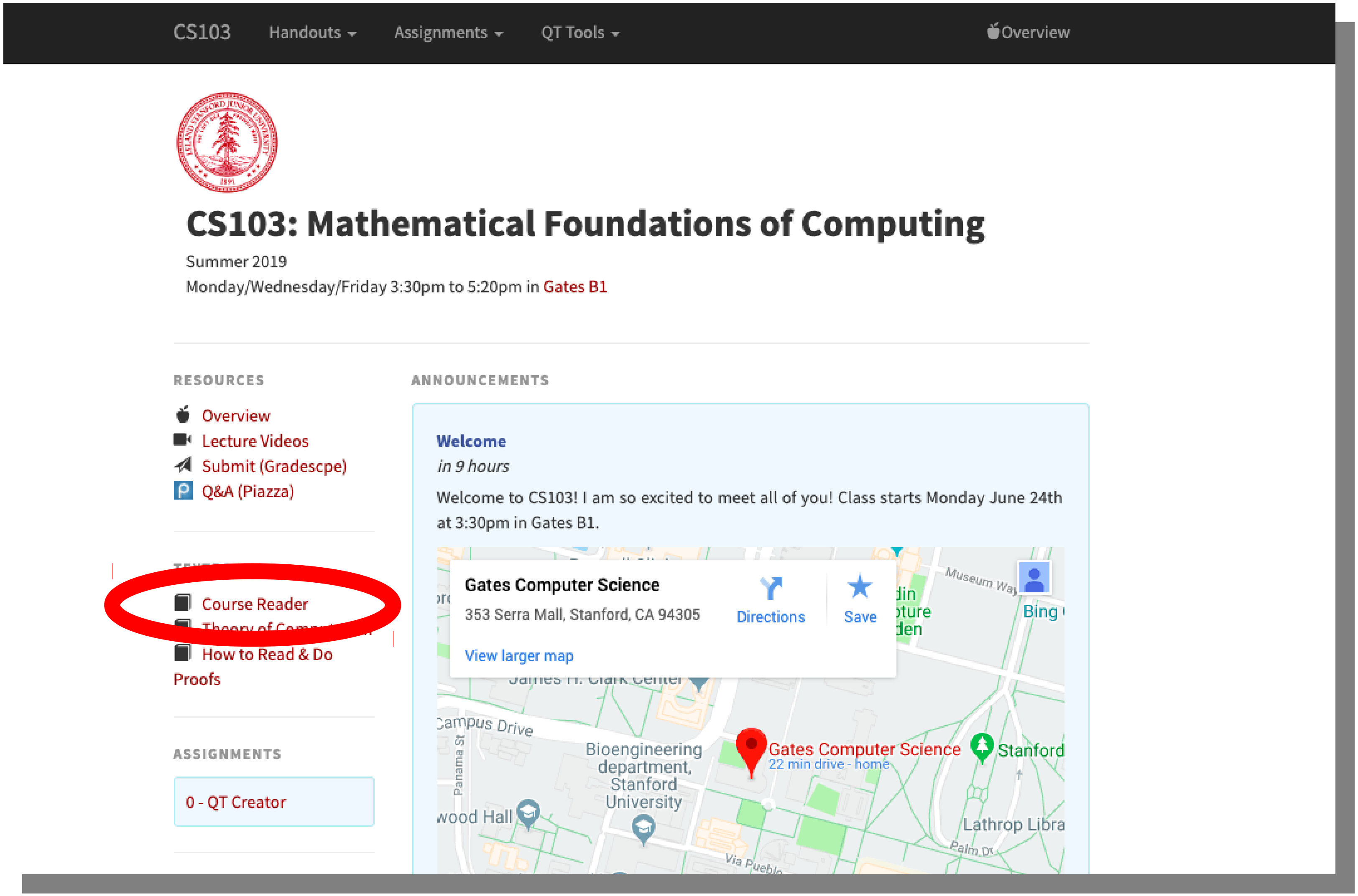
- Your first assignment, Problem Set 0, goes out today. It's due Friday at 3:00PM.
  - You'll need to get your development environment set up, though there's no actual coding involved.
  - We hope you have fun with this one – you'll learn some cool party tricks as you work through the assignment. ☺

Recommended Reading



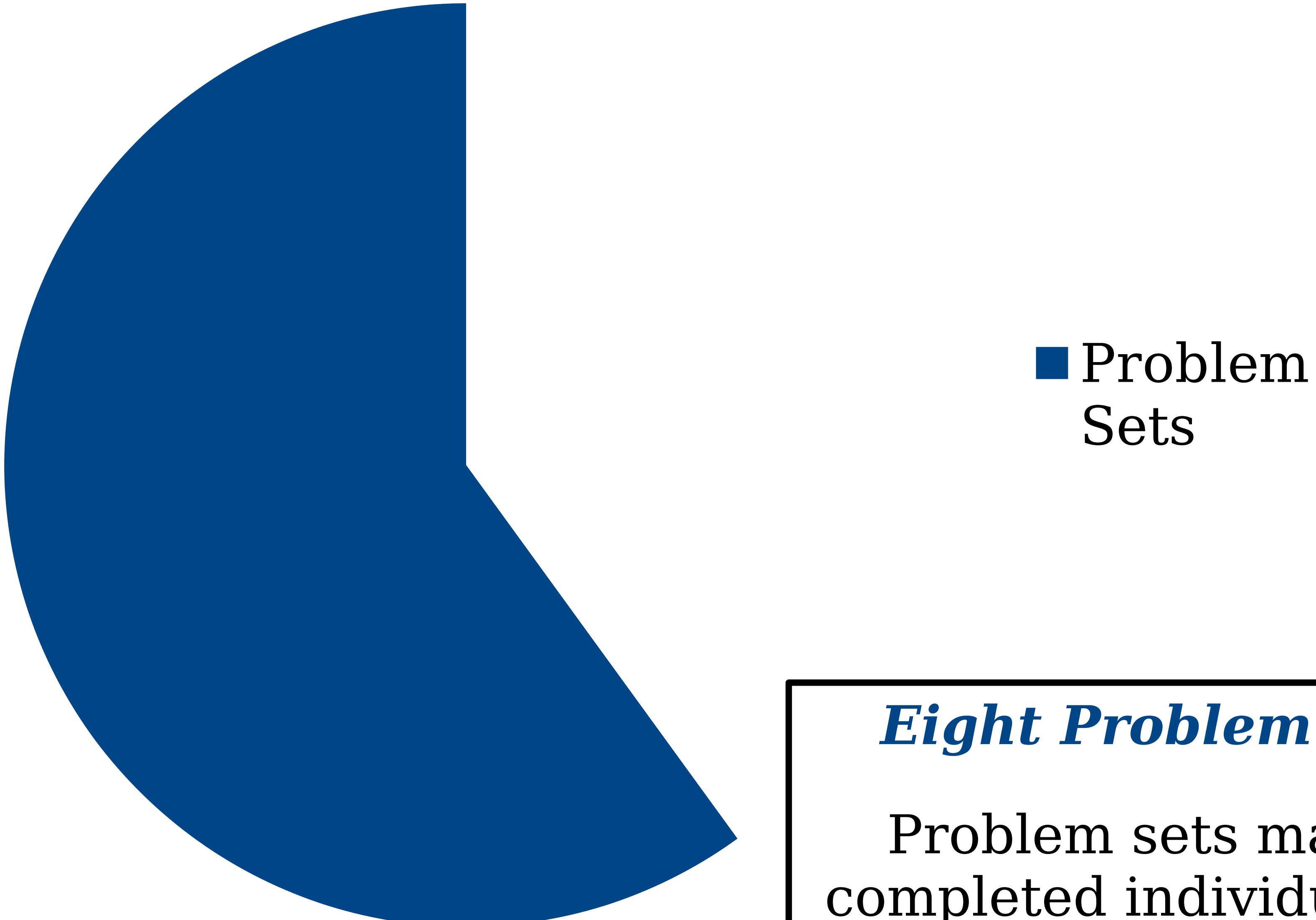


Online Course Notes



Grading

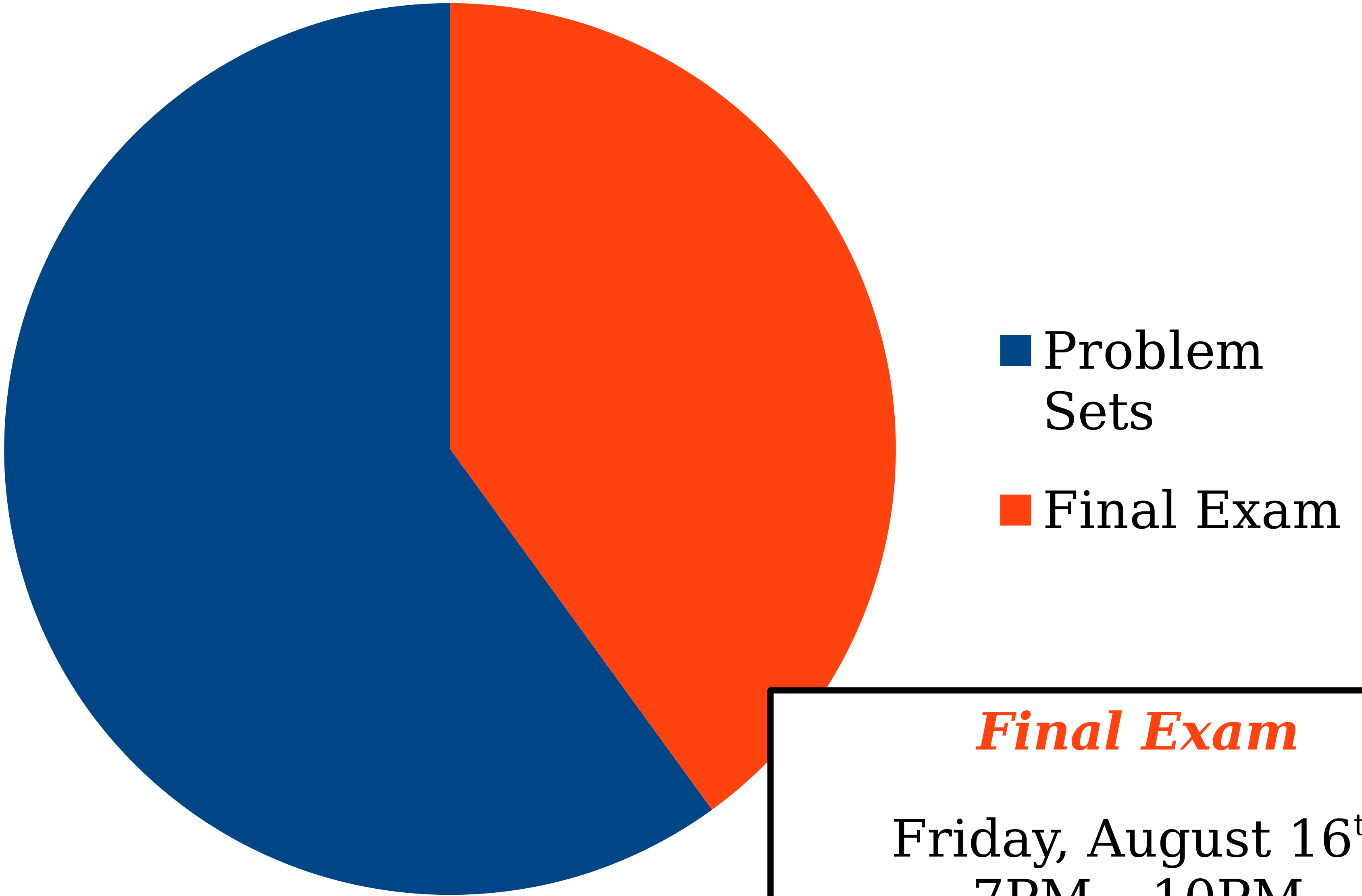
Grading



Eight Problem Sets

Problem sets may be completed individually or in pairs.

Grading



Final Exam

Friday, August 16th
7PM – 10PM

How to Succeed in CS103

Proof-Based Mathematics

- Most high-school math classes – with the exception of geometry – focus on *calculation*.
- CS103 focuses on *argumentation*.
- Your goal is to *see why things are true,* not *check that they work in a few cases*.
- Be curious! Ask questions. Try things out on your own. You'll learn this material best if you engage with it and refuse to settle for a “good enough” understanding.

Mental Traps to Avoid

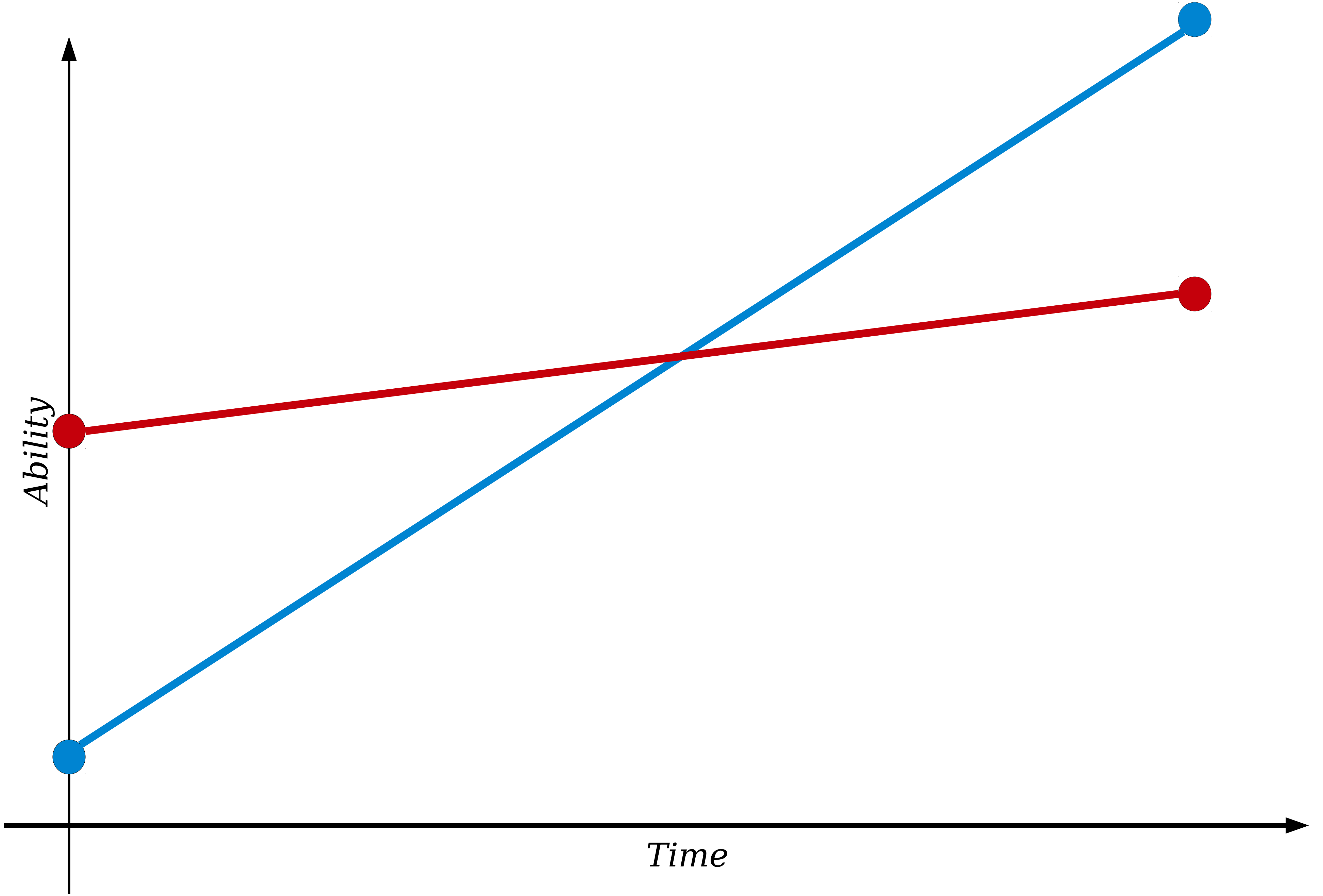
- “Everyone else has been doing math since before they were born and there is no way I'll ever be as good as them.”
- “A small minority of people are math geniuses and everyone else has no chance at being good at math.”
- “Being good at math means being able to instantly solve any math problem thrown at you.”

Mental Traps to Avoid

- “Everyone else has been doing math since before they were born and there is no way I'll ever be as good as them.”

  “A small minority of people are math geniuses and everyone else has no chance at being good at math.”

  “Being good at math means being able to instantly solve any math problem thrown at you.”



“A little slope makes up for a lot of *y*-intercept.”
- John Ousterhout

Mental Traps to Avoid

- “Everyone else has been doing math since before they were born and there is no way I'll ever be as good as them.”
- “A small minority of people are math geniuses and everyone else has no chance at being good at math.”
- “Being good at math means being able to instantly solve any math problem thrown at you.”

Mental Traps to Avoid

- "Everyone else has been doing math since before they were born and there is no way I'll ever be as good as them."
- "A small minority of people are math geniuses and everyone else has no chance at being good at math."
- "Being good at math means being able to instantly solve any math problem thrown at you."

Pro Tip #1:

Never Confuse Experience for Talent

Pro Tip #2:

Have a Growth Mindset

Fun Math Question

Suppose you improve at some skill at a rate of 1% per day. How much better at that skill will you be by the end of the year?

After one day, you're 1.01 times better.
After two days, you're $(1.01)^2$ times better.

After one year, you'll be
$(1.01)^{365} \approx 37.8$ times better!

Mental Traps to Avoid

- “Everyone else has been doing math since before they were born and there is no way I'll ever be as good as them.”
- “A small minority of people are math geniuses and everyone else has no chance at being good at math.”
- “Being good at math means being able to instantly solve any math problem thrown at you.”

Mental Traps to Avoid

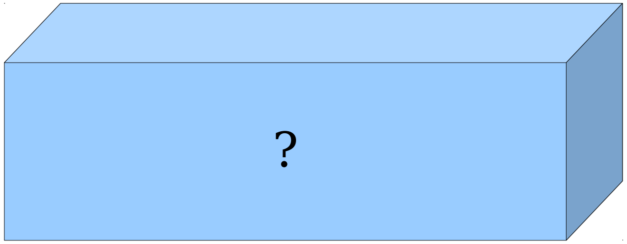
“Everyone else has been doing math since before they were born and there is no way I'll ever be as good as them.”

“A small minority of people are math geniuses and everyone else has no chance at being good at math.”

- “Being good at math means being able to instantly solve any math problem thrown at you.”

Simple Open Problems

- Math is often driven by seemingly simple problems that no one knows the answer to.
- Example: the ***integer brick problem***:



Is there a rectangular brick where any line connecting two corners has integer length?

- Having open problems like these drives the field forward – it motivates people to find new discoveries and to invent new techniques.

My Advice

- Question everything!
- Come to lecture :)
- Study strategically and intentionally
- Stay on top of the material and actively patch any holes in your understanding
- Persevere, but know when to get help

We've got a big journey ahead of us.

Let's get started!

“CS103 students”

“All the computers on the Stanford network”

“Cool people”

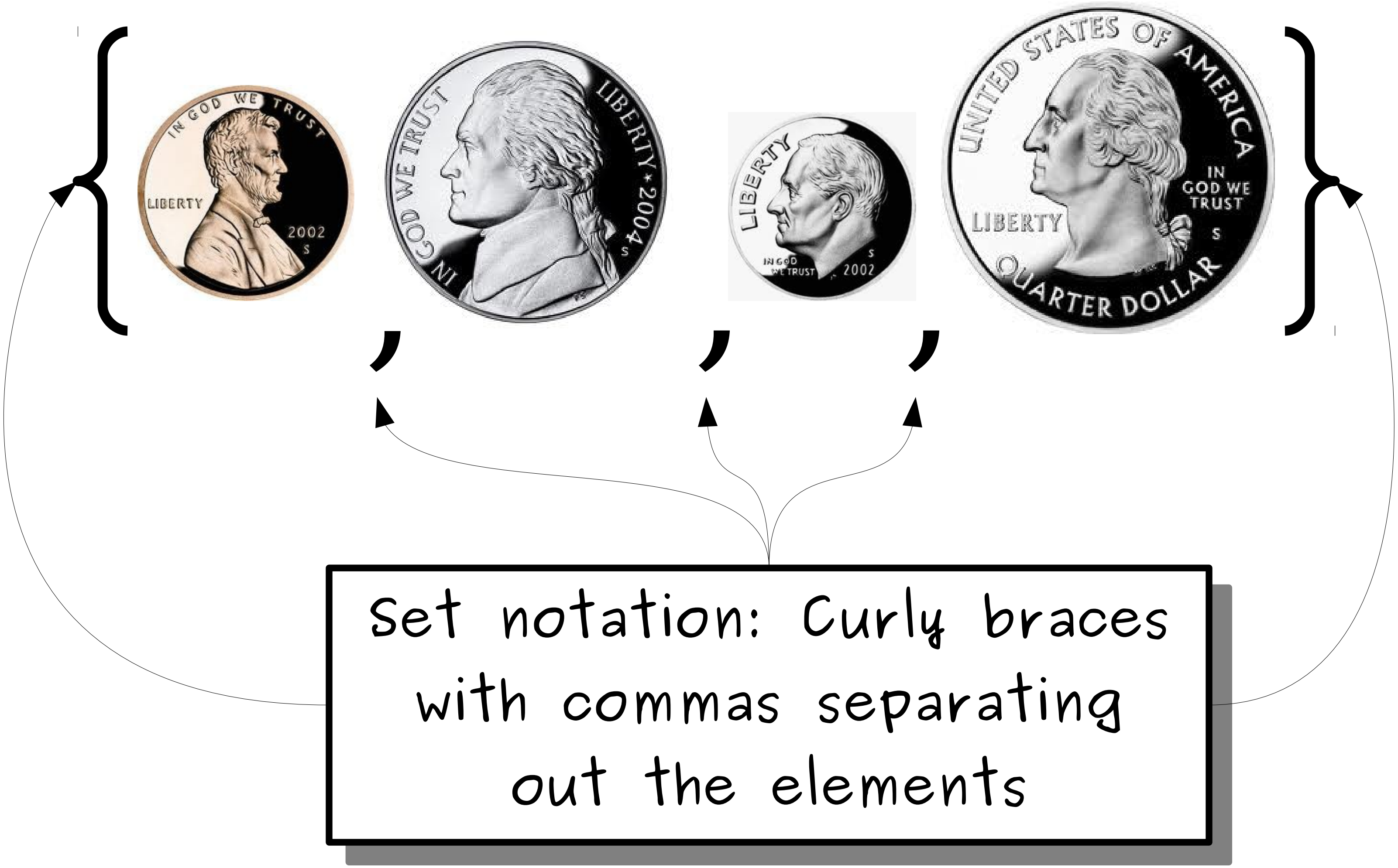
“The chemical elements”

“Cute animals”

“US coins”

A ***set*** is an unordered collection of distinct objects, which may be anything (including other sets).

A ***set*** is an unordered collection of distinct objects, which may be anything (including other sets).



A ***set*** is an unordered collection of distinct objects, which may be anything (including other sets).

Two sets are equal when they have exactly the same contents, ignoring order.

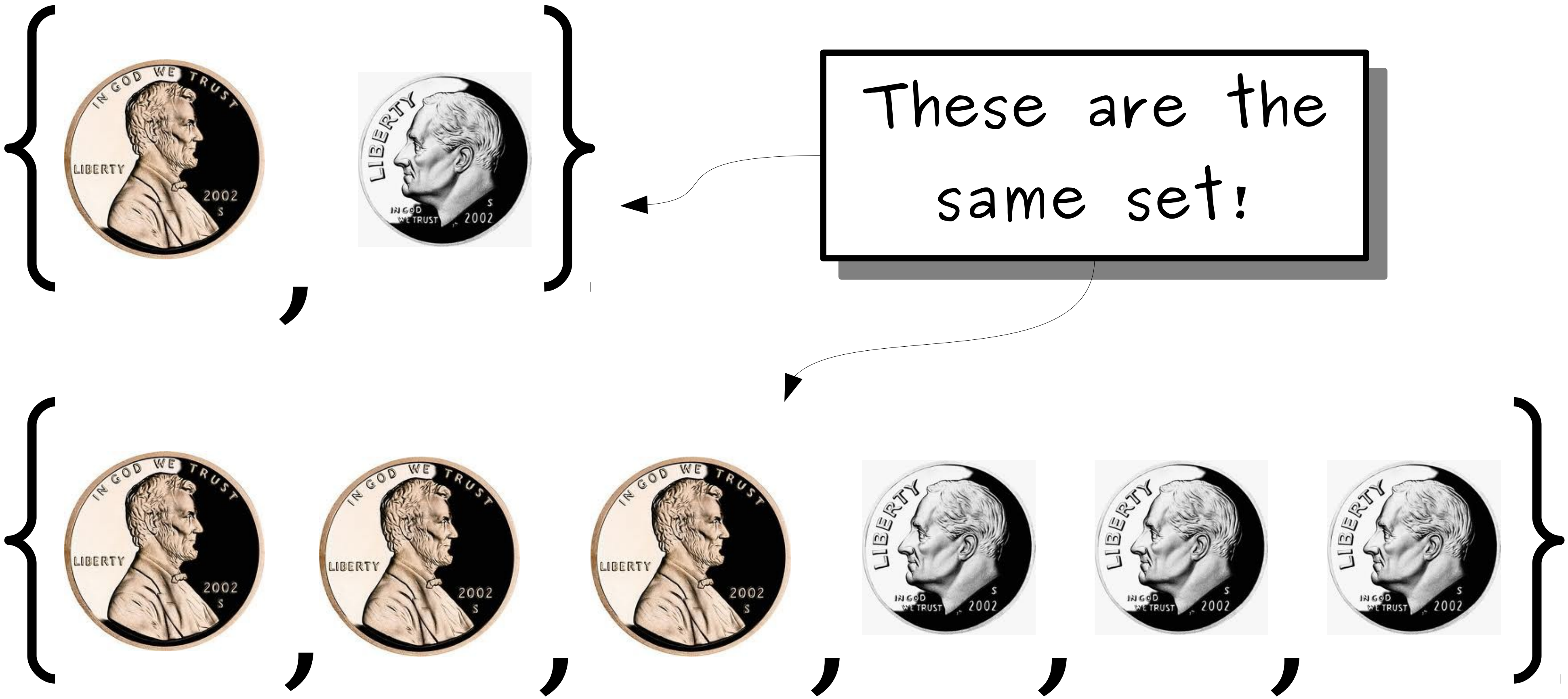
Two sets are equal when they have exactly the same contents, ignoring order.

Two sets are equal when they have exactly the same contents, ignoring order.

Sets cannot contain the same object twice.
Repeated elements are ignored.

Sets cannot contain the same object twice.
Repeated elements are ignored.

Sets cannot contain the same object twice.
Repeated elements are ignored.



Sets cannot contain the same object twice.
Repeated elements are ignored.

$$\{\ \} = \emptyset$$

The empty set contains no elements.

We use this symbol to denote the empty set.

This is a number.

This is a set. It contains a number.

$$1 \stackrel{?}{=} \{1\}$$

Are these objects equal to one another?

This is a number.

This is a set. It contains a number.

$$1 \neq \{1\}$$

Are these objects equal to one another?

This set contains nothing at all.

This set has one element, which happens to be the empty set.

$$\emptyset \stackrel{?}{=} \{\emptyset\}$$

Are these objects equal to one another?

This set contains nothing at all.

This set has one element, which happens to be the empty set.

$$\emptyset \neq \{\emptyset\}$$

Are these objects equal to one another?

Membership

Membership

{ 
IN GOD WE TRUST
LIBERTY
2004
S
, 
IN GOD WE TRUST
LIBERTY
2002
S
, 
UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
S
QUARTER DOLLAR
, 
LIBERTY
IN GOD WE TRUST
S
2002
}

Membership

Is  in this set?

Membership

∈ {


,


,


,


}

Is


in this set?

Membership

∈ {

,

,

,

}

Is

in this set?

Membership

∉ {

,

,

,

}

Is

in this set?

Set Membership

- Given a set S and an object x, we write

  $$\boldsymbol{x \in S}$$

  if x is contained in S, and

  $$\boldsymbol{x \notin S}$$

  otherwise.
- If $x \in S$, we say that x is an ***element*** of S.
- Given any object x and any set S, either $x \in S$ or $x \notin S$.

Infinite Sets

- Some sets contain *infinitely many* elements!
- The set ℕ = { 0, 1, 2, 3, …} is the set of all the ***natural numbers***.
    - Some mathematicians don't include zero; in this class, assume that 0 is a natural number.
- The set ℤ = { …, -2, -1, 0, 1, 2, … } is the set of all the ***integers***.
    - Z is from German "Zahlen."
- The set ℝ is the set of all ***real numbers***.
    - $e \in \mathbb{R}$, $\pi \in \mathbb{R}$, $4 \in \mathbb{R}$, etc.

Describing Complex Sets

- Here are some English descriptions of infinite sets:

  "The set of all even natural numbers."

  "The set of all real numbers less than 137."

  "The set of all negative integers."

- To describe complex sets like these mathematically, we'll use ***set-builder notation***.

Even Natural Numbers

$$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$$

Even Natural Numbers

$$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$$

Even Natural Numbers

$$\{ \; n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \; \}$$

The set of all n

Even Natural Numbers

$$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$$

The set of all n

where

Even Natural Numbers

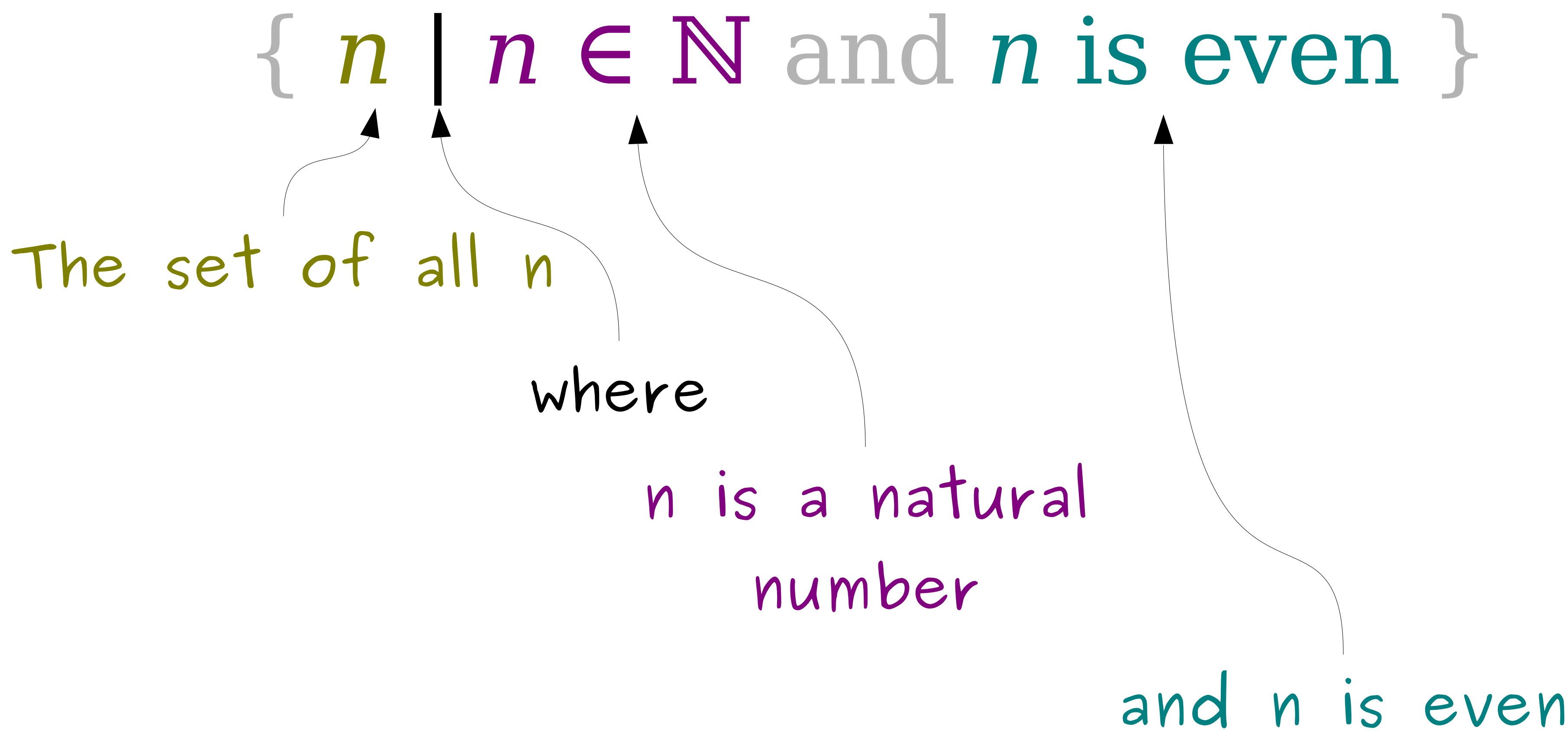
$$\{\ n \mid n \in \mathbb{N} \text{ and } n \text{ is even } \}$$

The set of all n

where

n is a natural number

Even Natural Numbers



Even Natural Numbers

$$\{\ n \mid n \in \mathbb{N} \text{ and } n \text{ is even } \}$$

The set of all n

where

n is a natural number

and n is even

$$\{\ 0, 2, 4, 6, 8, 10, 12, 14, 16, \ldots\ \}$$

Set Builder Notation

- A set may be specified in ***set-builder notation***:

$$\{\ x \mid \textit{some property } x \textit{ satisfies}\ \}$$

- For example:

$\{\ r \mid r \in \mathbb{R} \text{ and } r < 137\ \}$

$\{\ n \mid n \text{ is an even natural number}\ \}$

$\{\ S \mid S \text{ is a set of US currency}\ \}$

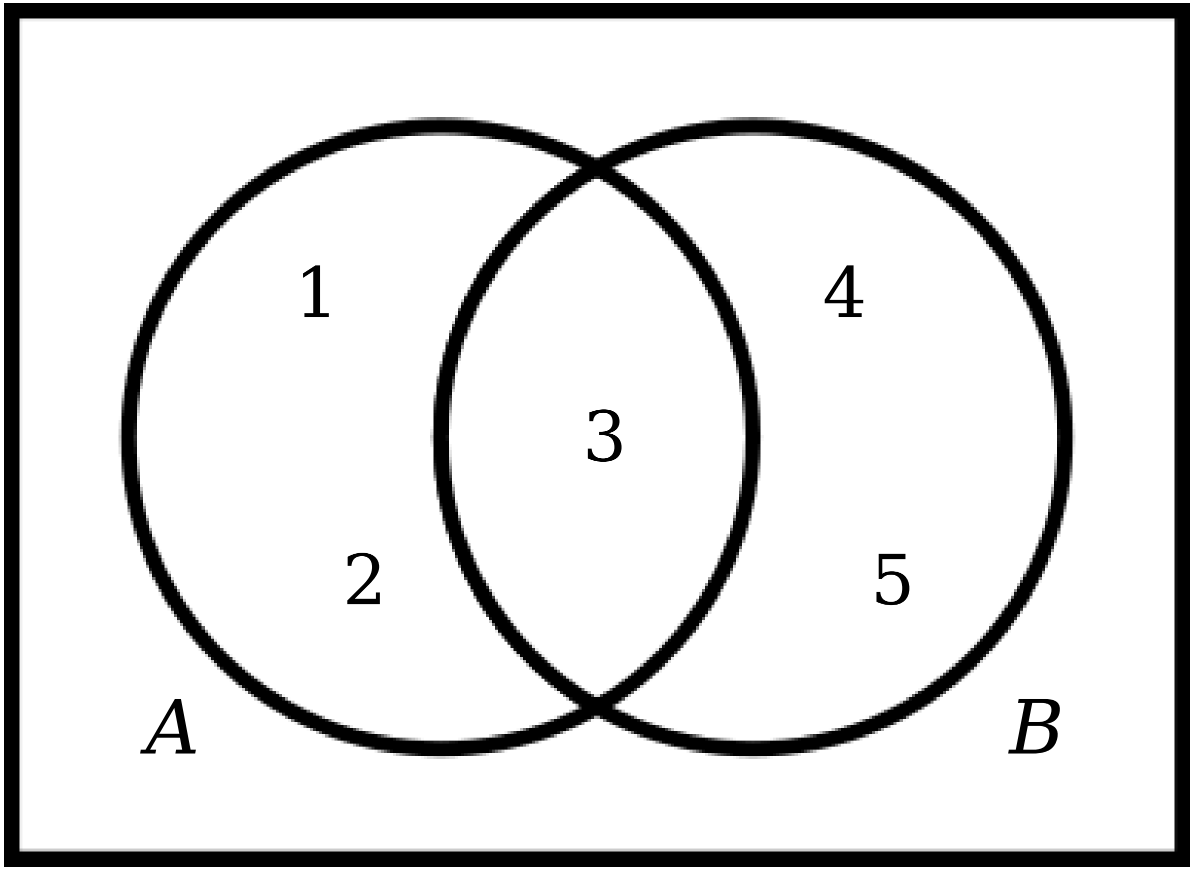
$\{\ a \mid a \text{ is cute animal}\ \}$

$\{\ r \in \mathbb{R} \mid r < 137\ \}$

$\{\ n \in \mathbb{N} \mid n \text{ is odd}\ \}$

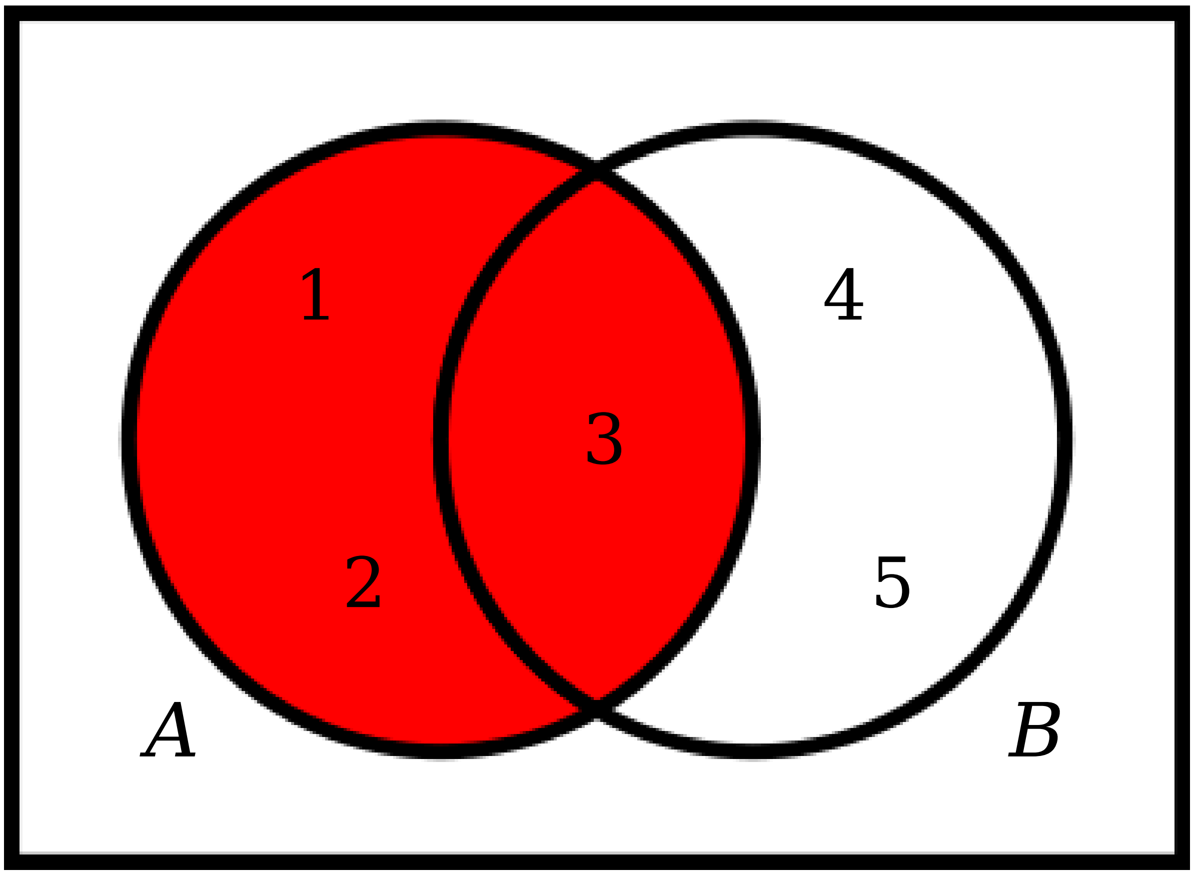
Combining Sets

Venn Diagrams



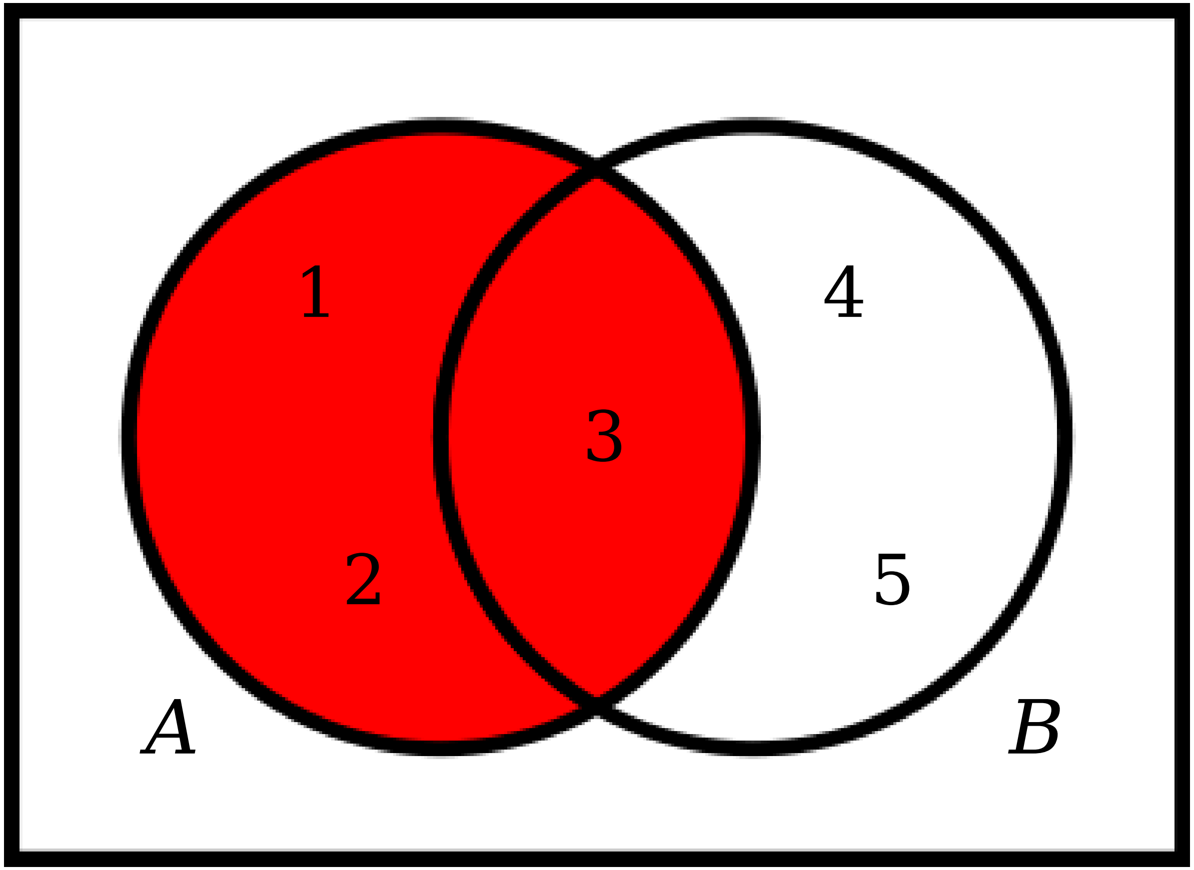
$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

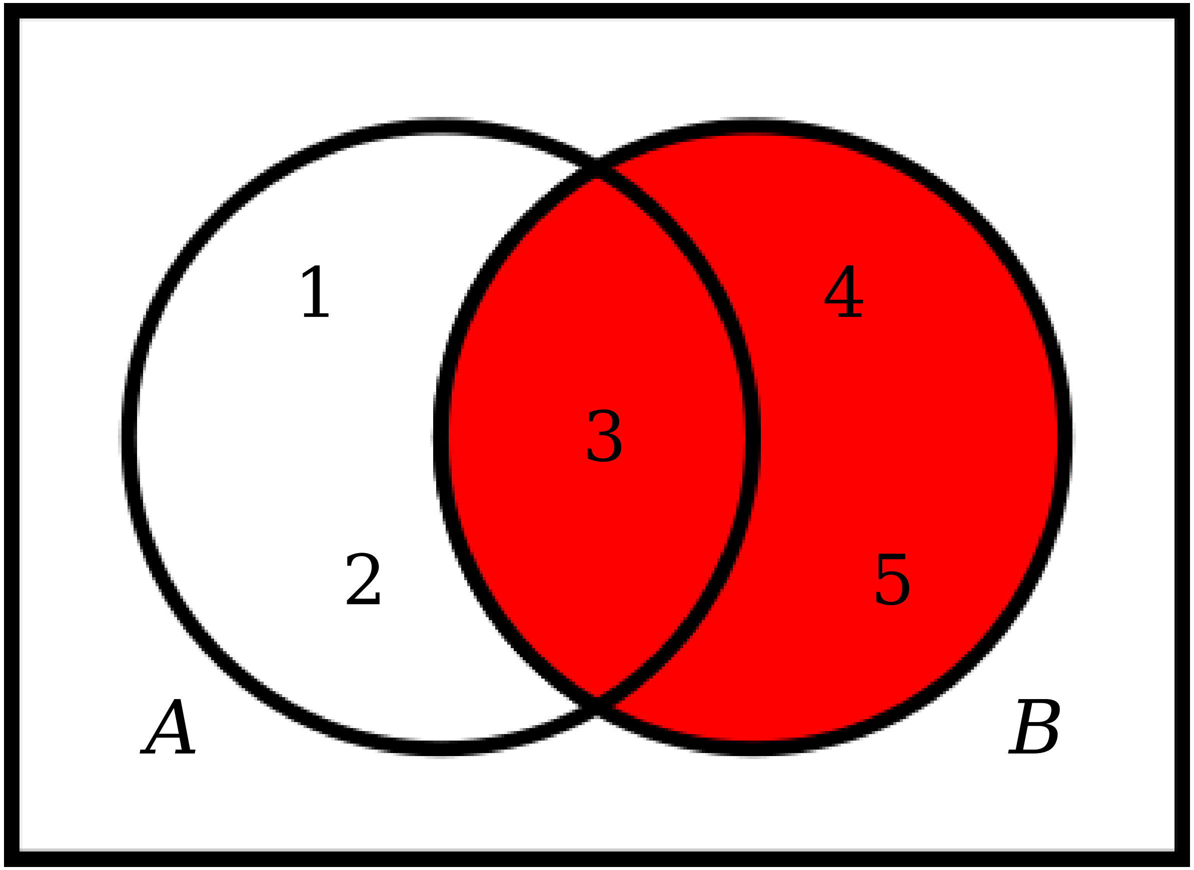
Venn Diagrams



A

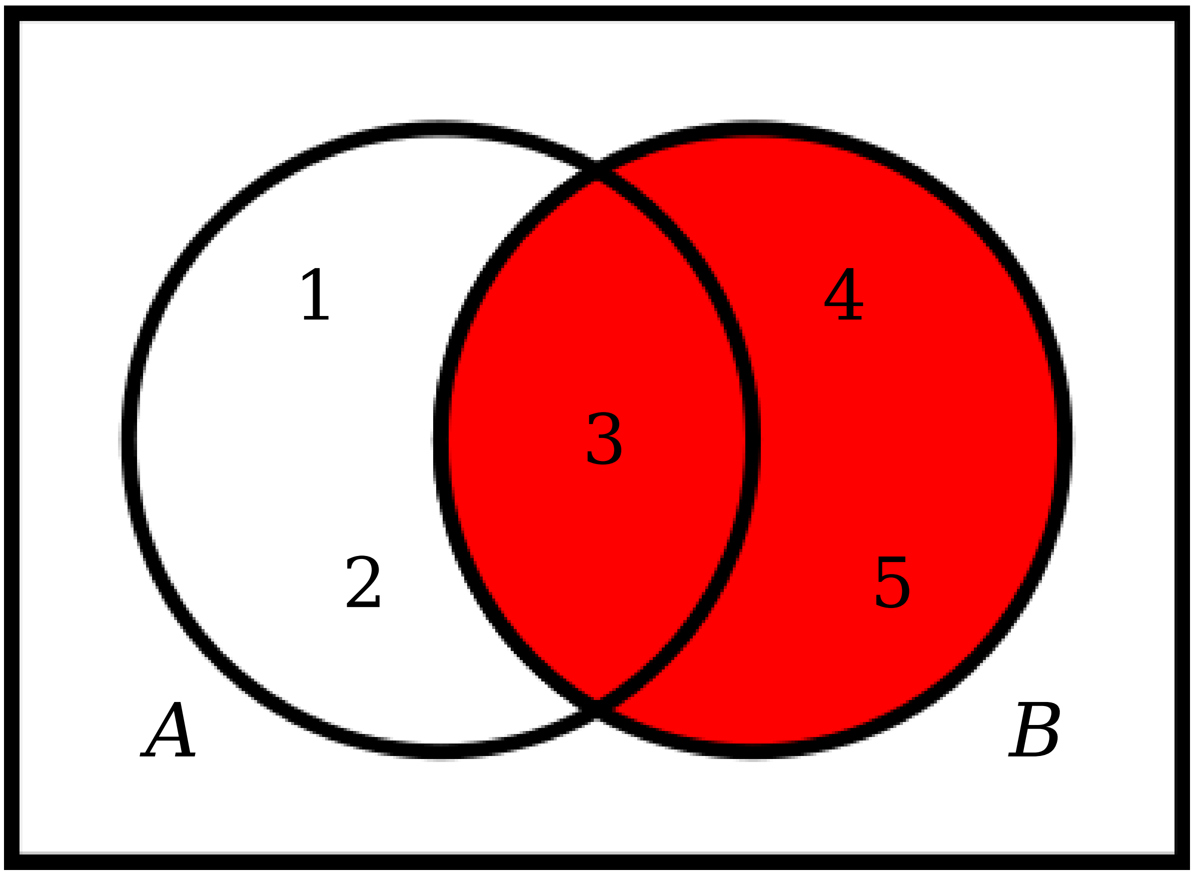
$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

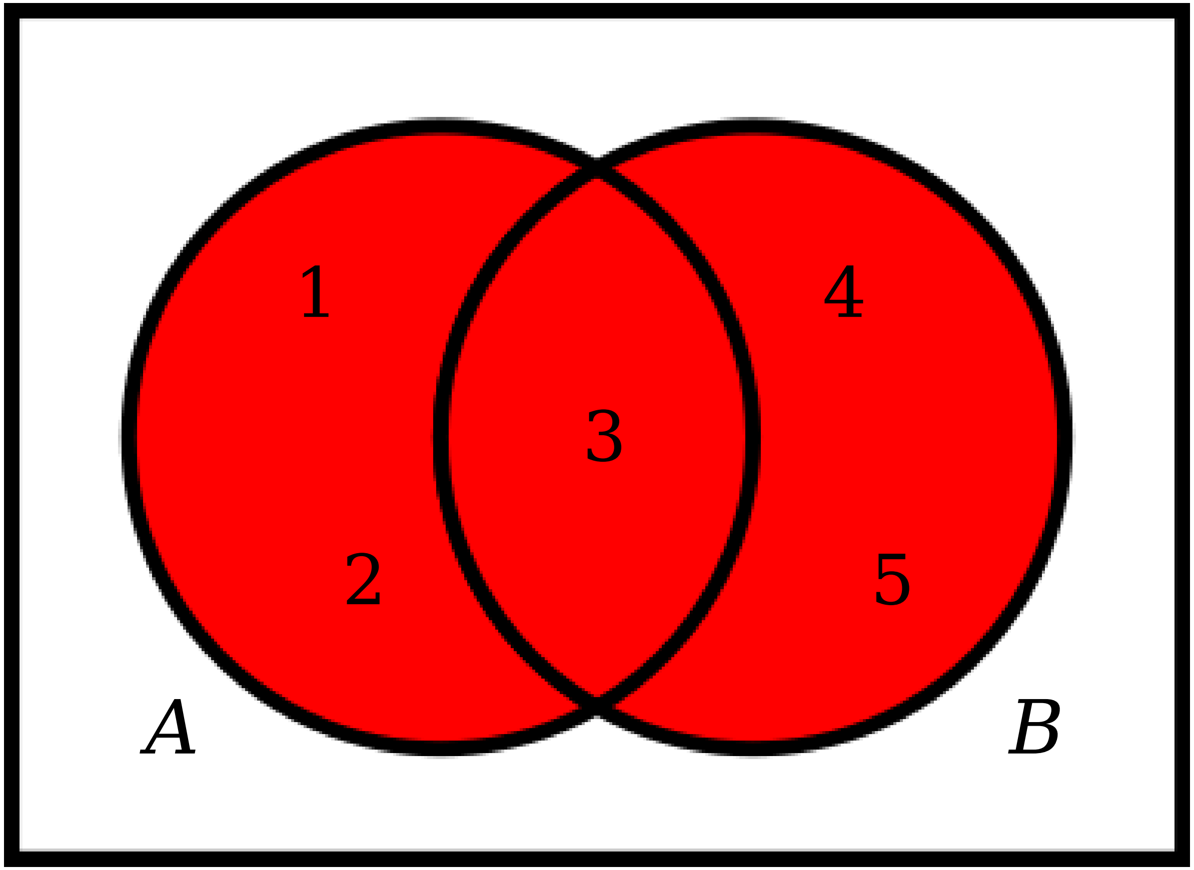
Venn Diagrams



B

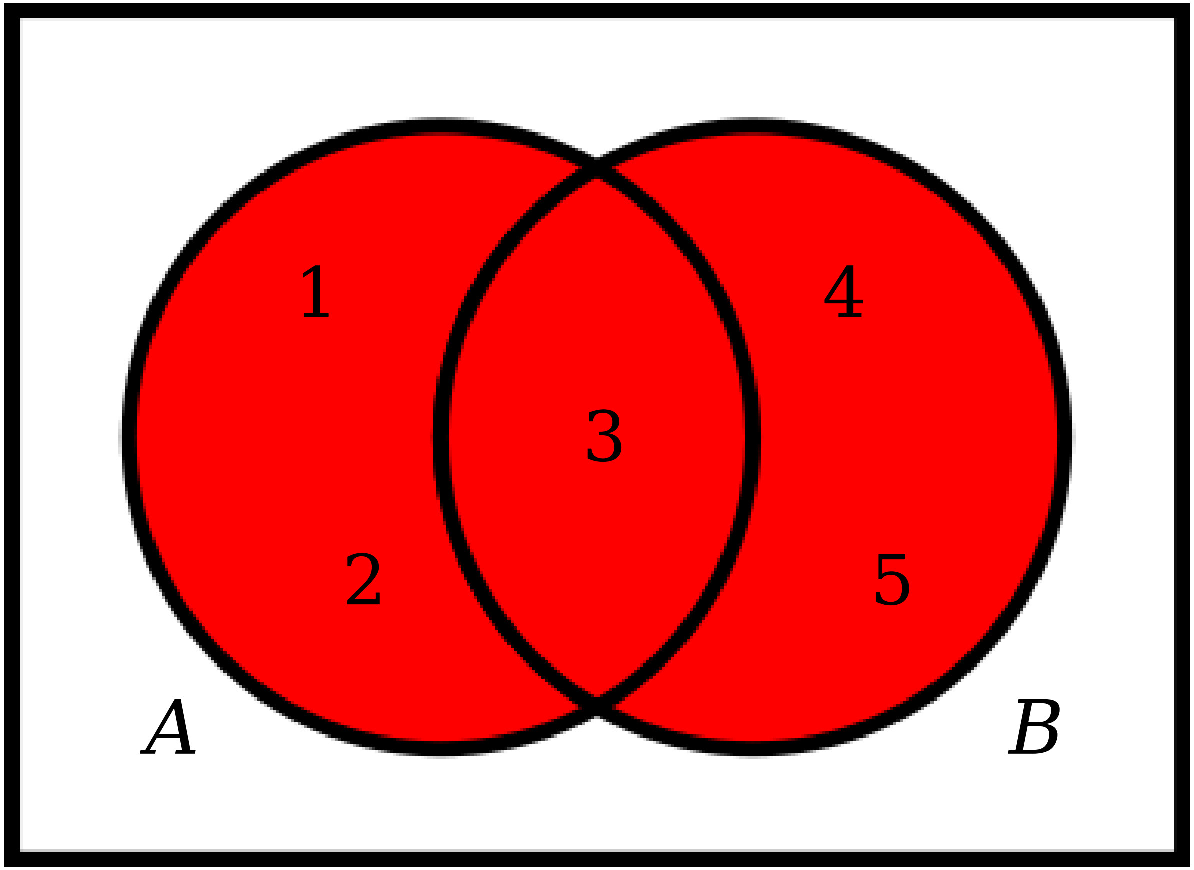
$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



Union

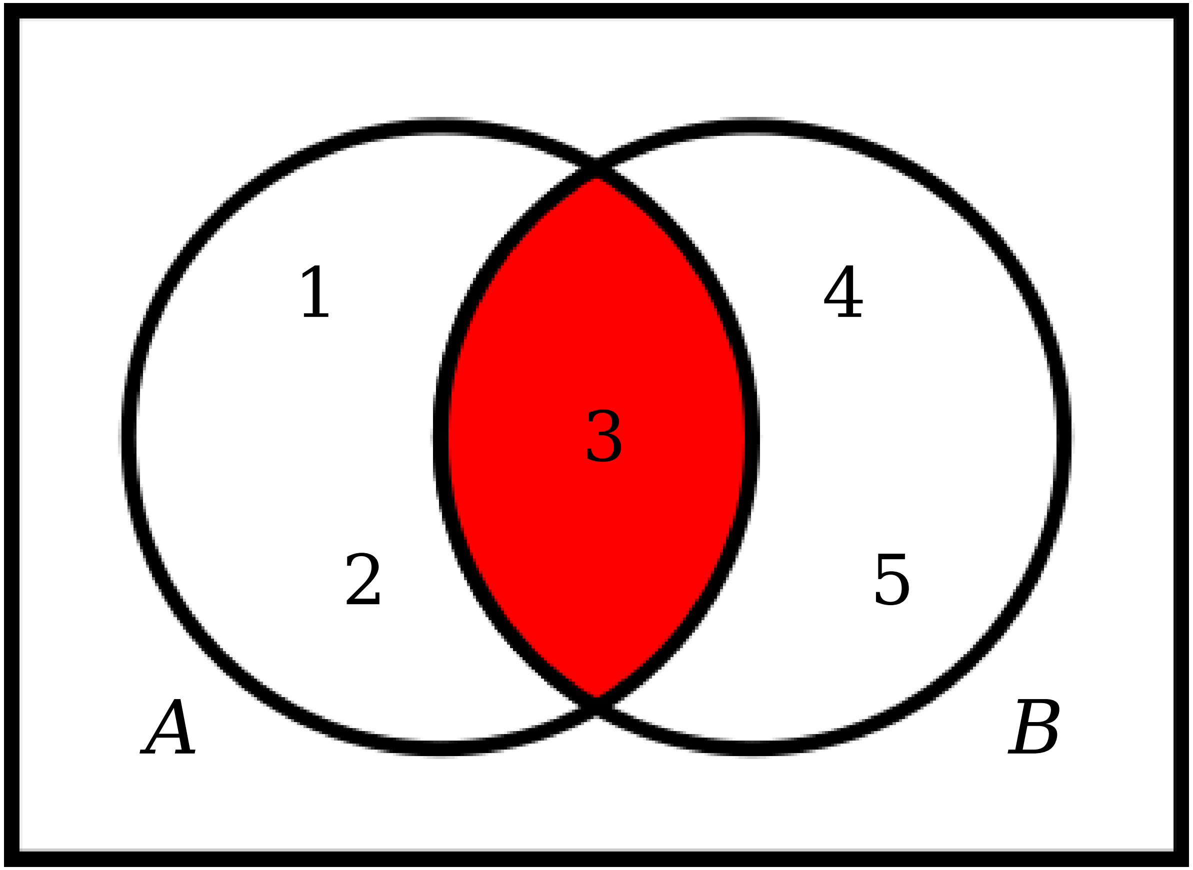
$A \cup B$

{ 1, 2, 3, 4, 5 }

$A = \{ 1, 2, 3 \}$

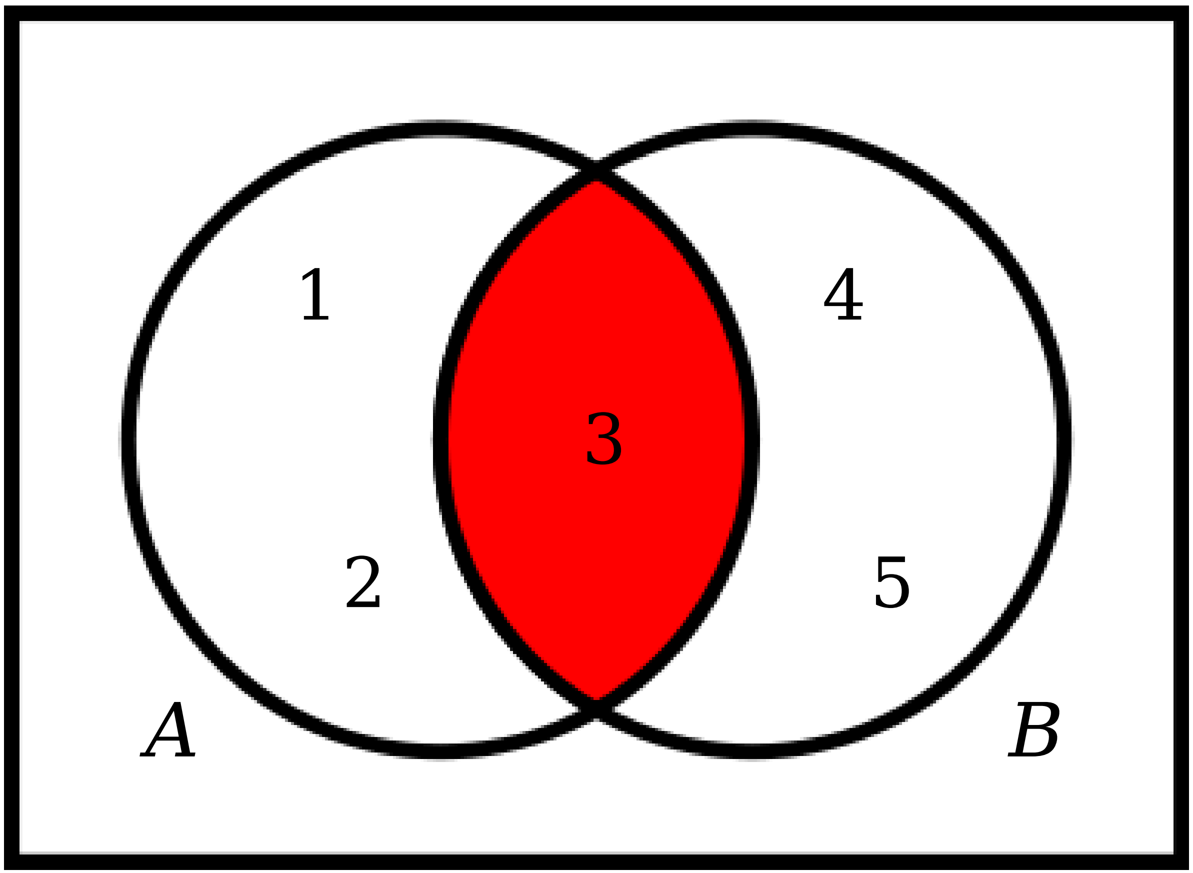
$B = \{ 3, 4, 5 \}$

Venn Diagrams



$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



Intersection

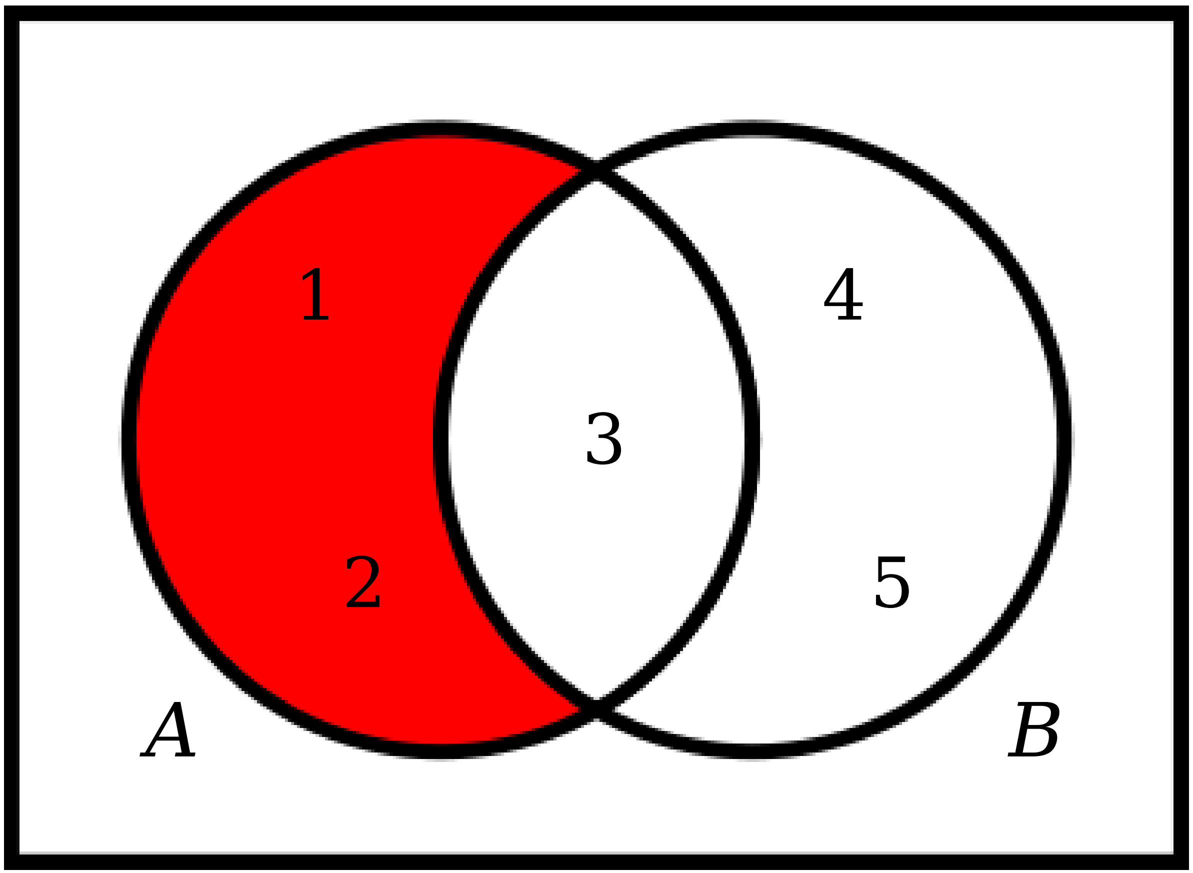
$A \cap B$

$\{ 3 \}$

$A = \{ 1, 2, 3 \}$

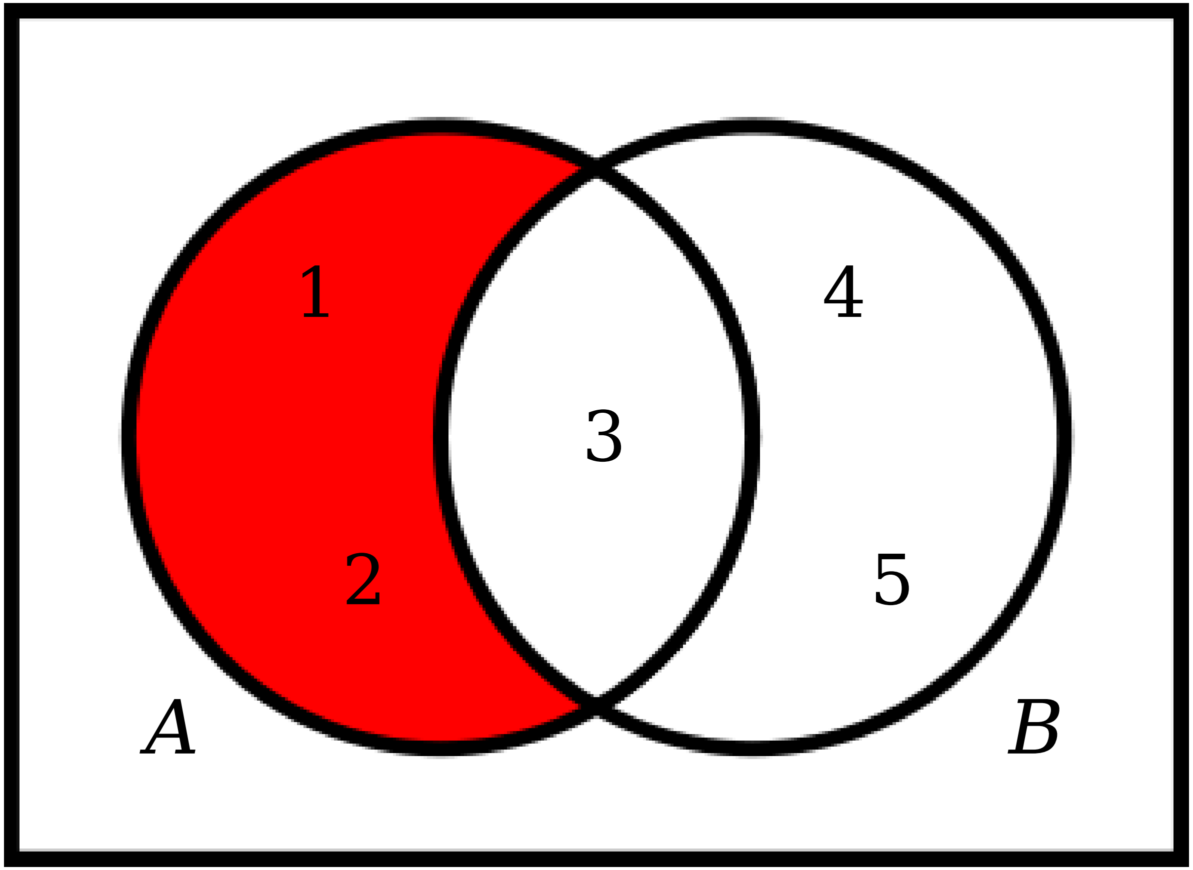
$B = \{ 3, 4, 5 \}$

Venn Diagrams



$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



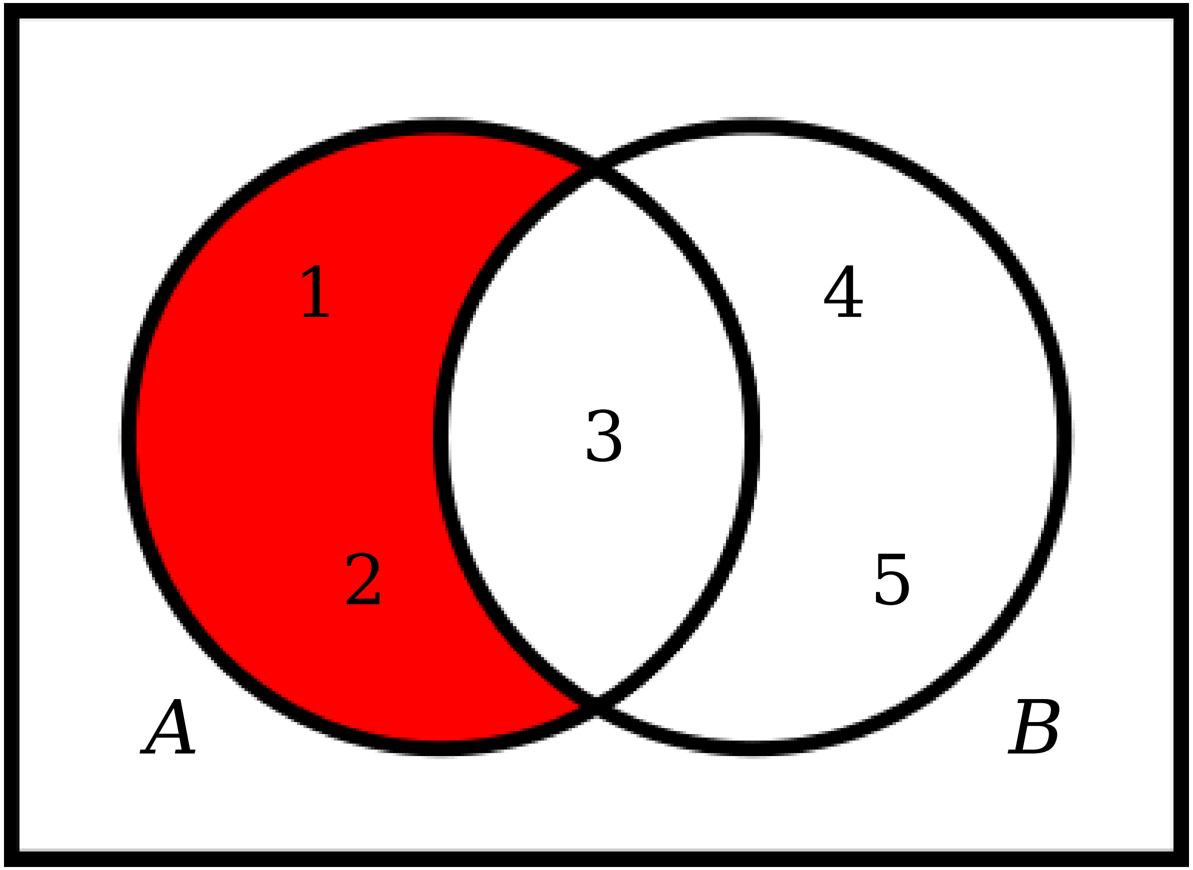
Difference

$A - B$

$\{ 1, 2 \}$

$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



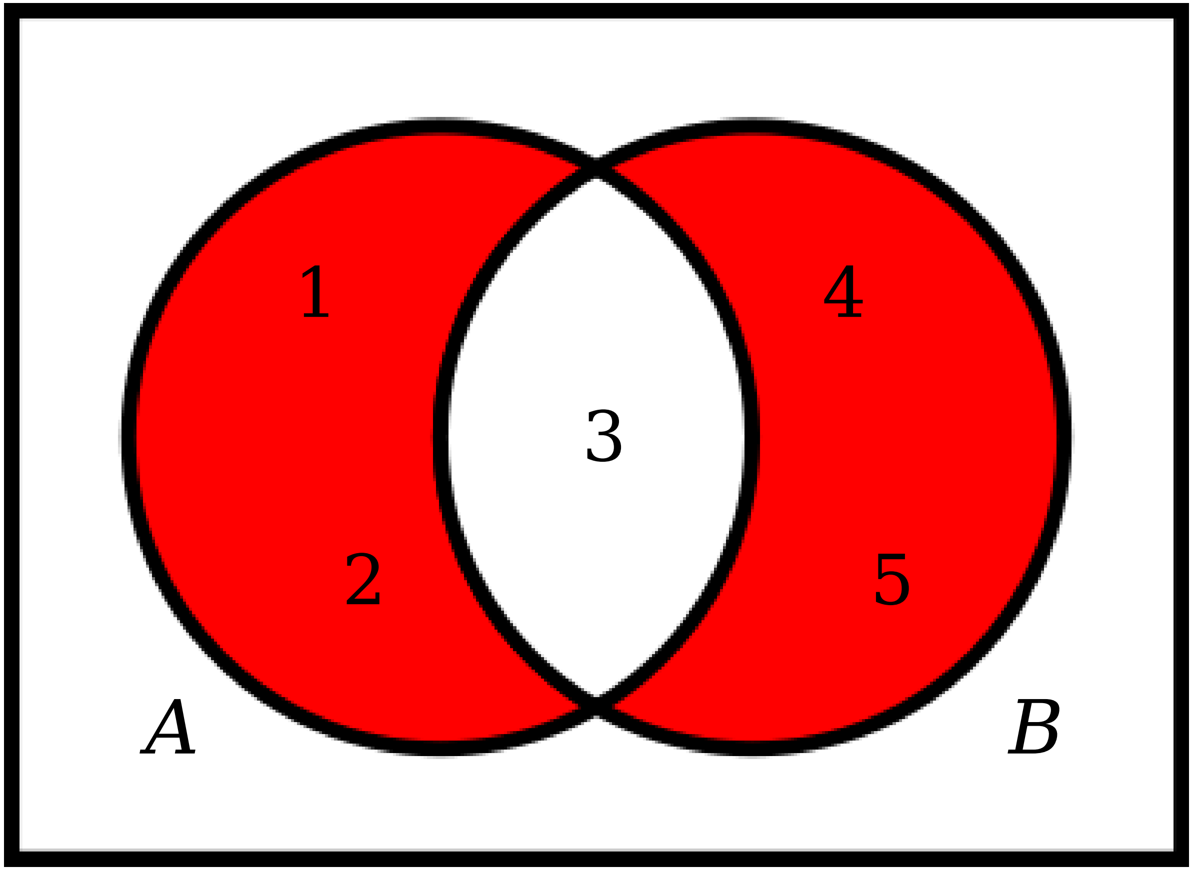
Difference

$A \setminus B$

{ 1, 2 }

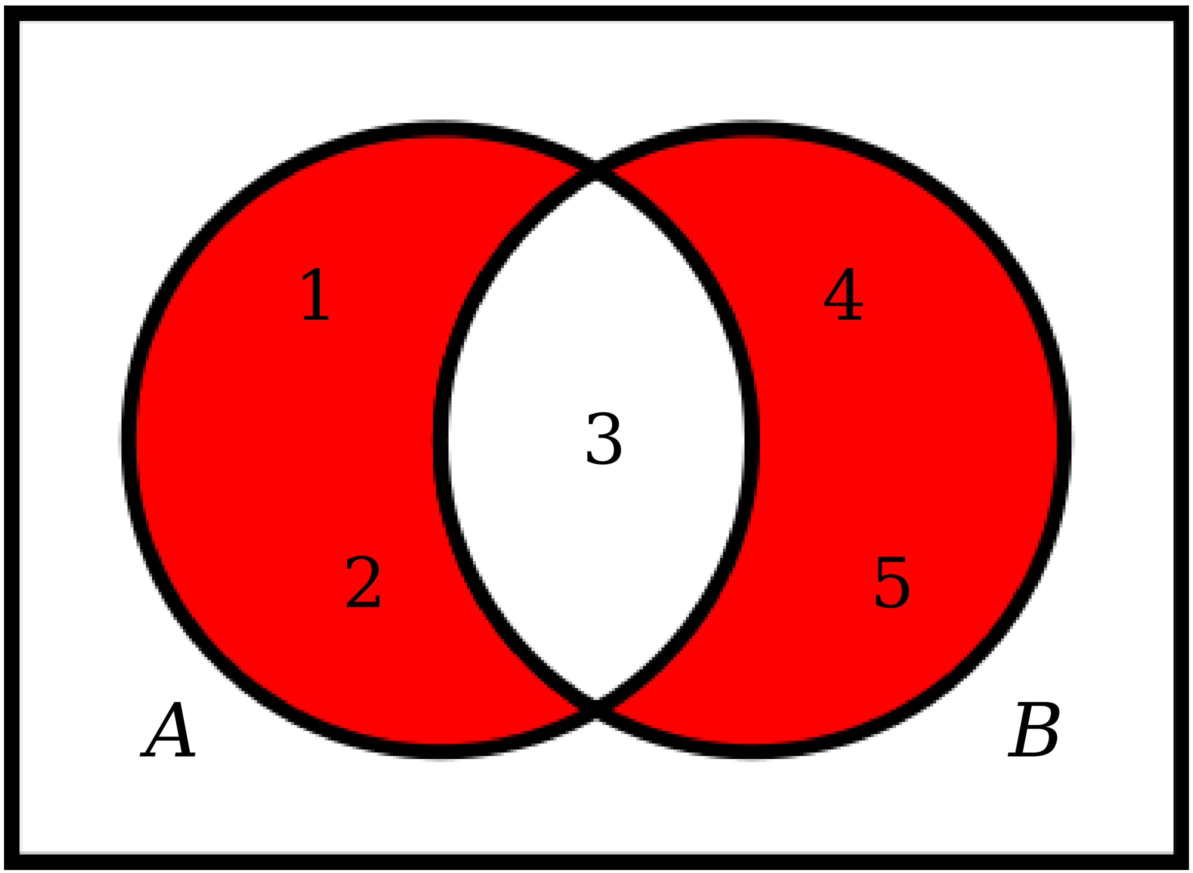
$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Venn Diagrams



Symmetric Difference

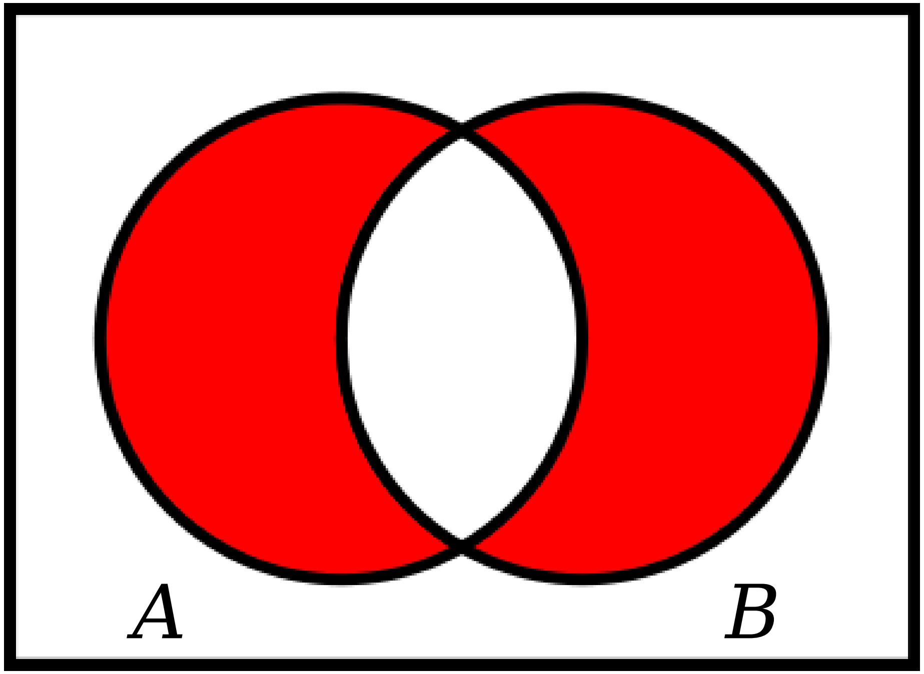
$A \, \Delta \, B$

$\{\ 1, 2, 4, 5\ \}$

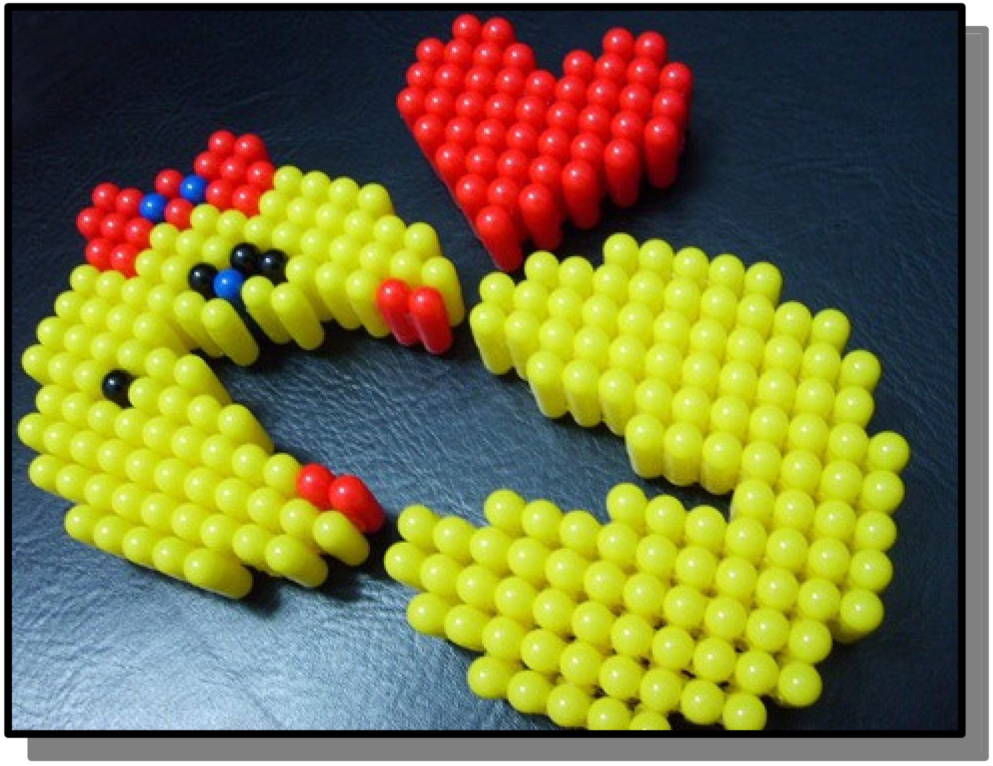
$A = \{\ 1, 2, 3\ \}$

$B = \{\ 3, 4, 5\ \}$

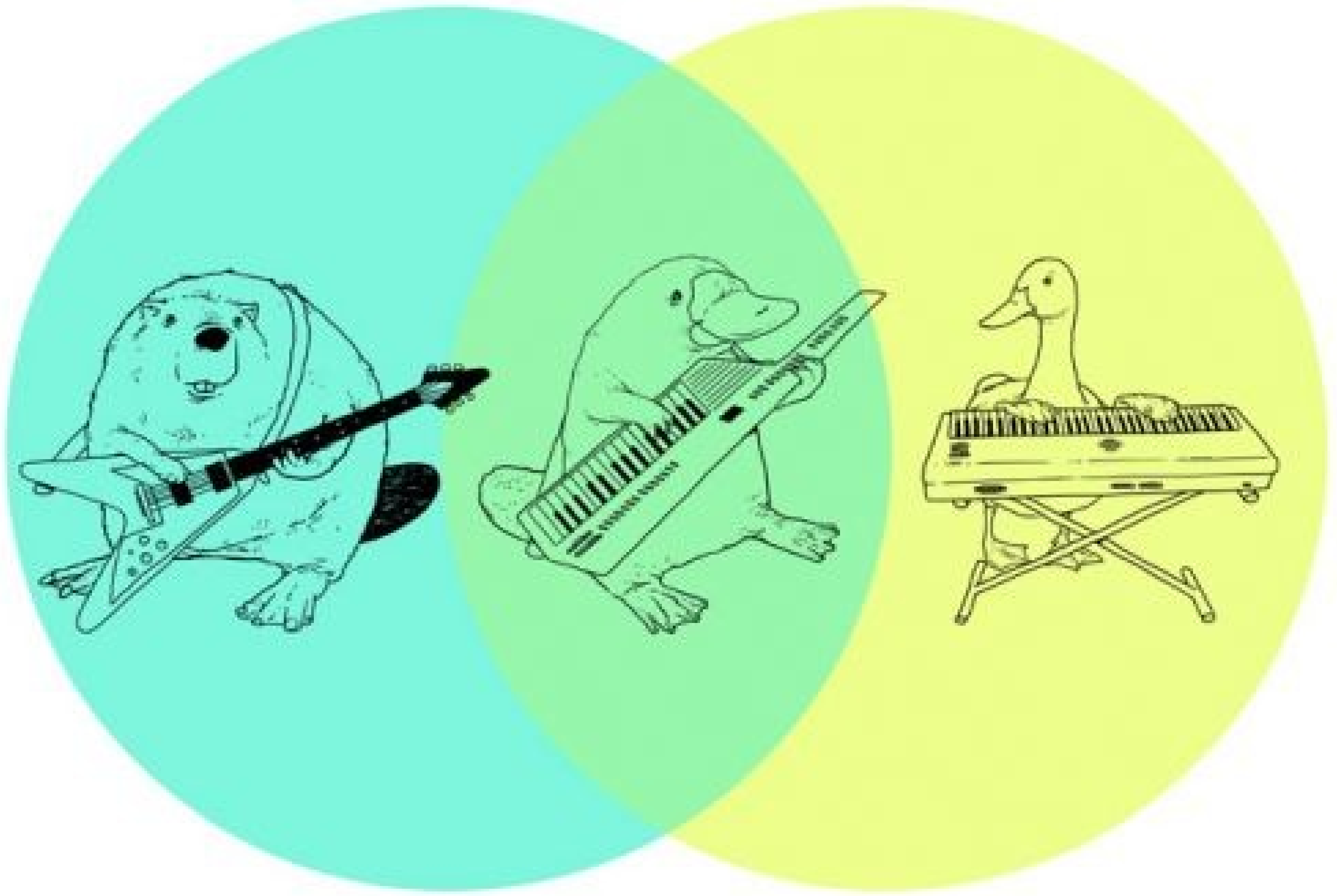
Venn Diagrams



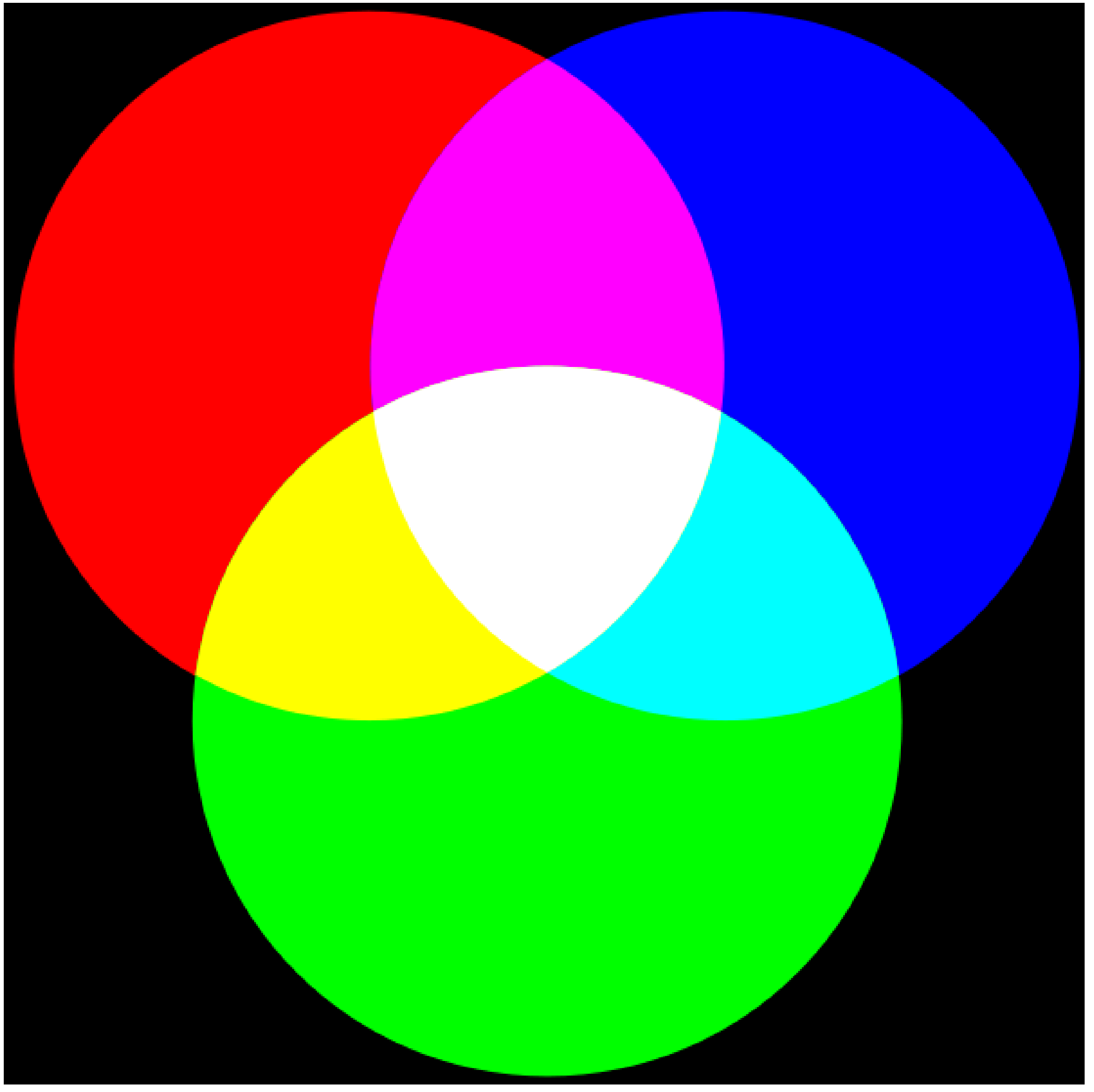
$A \, \Delta \, B$



Venn Diagrams

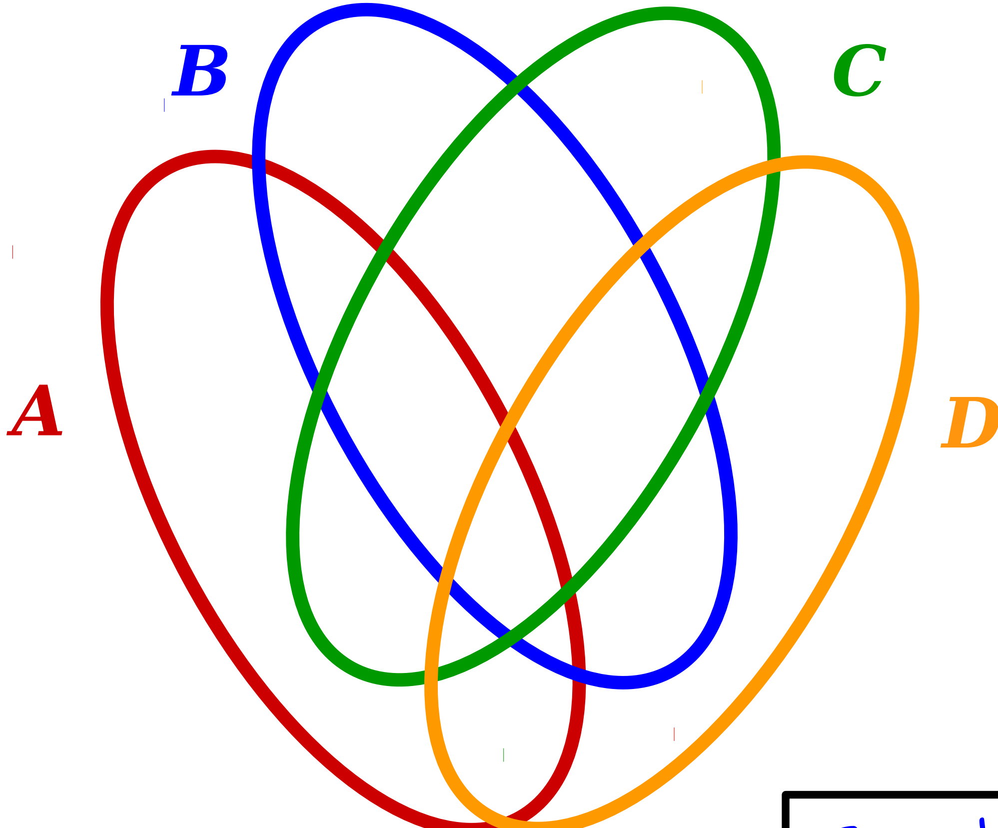


Venn Diagrams for Three Sets



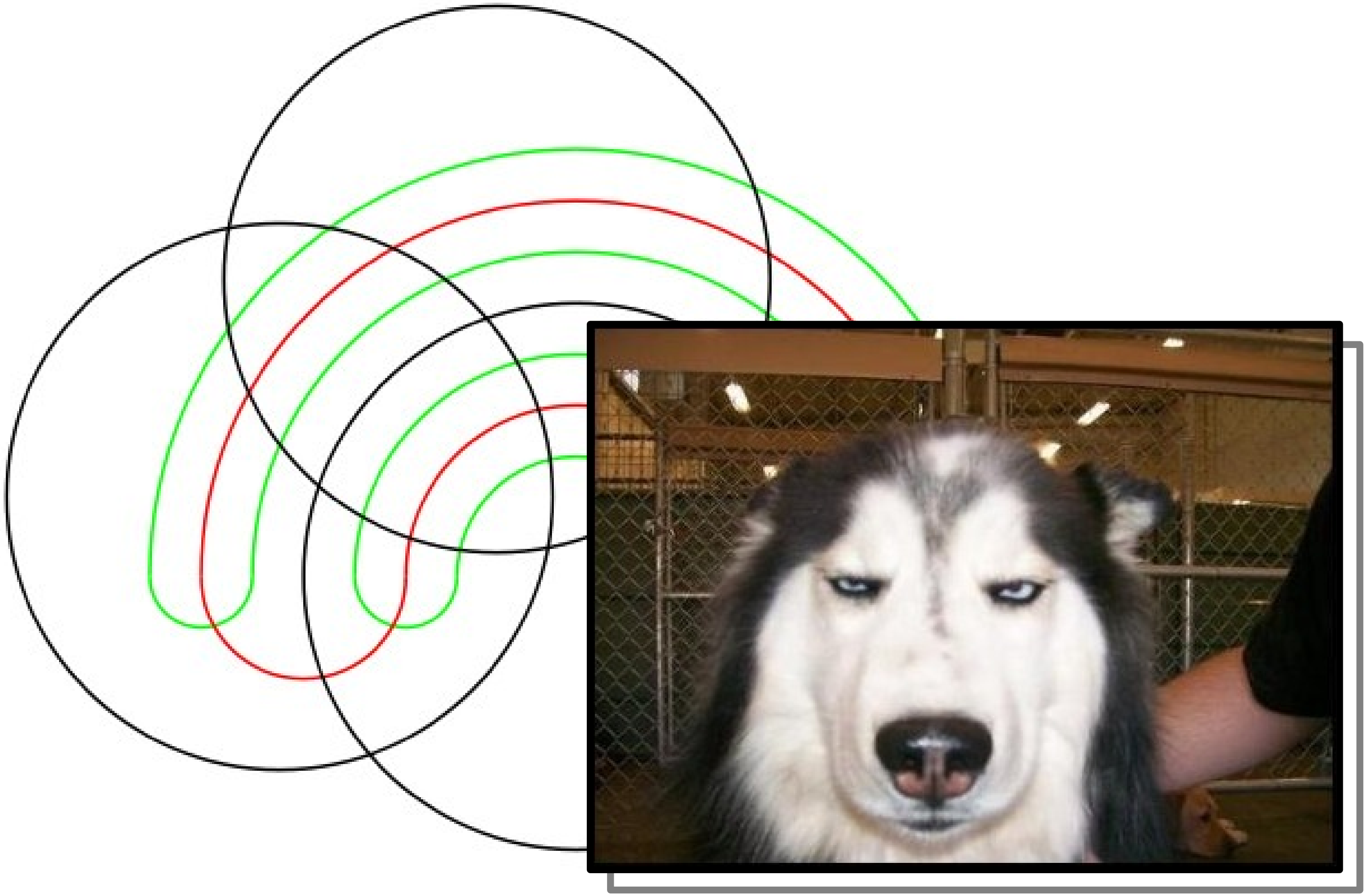
Venn Diagrams for Three Sets

Venn Diagrams for Four Sets



Question to ponder: why don't we just draw four circles?

Venn Diagrams for Five Sets



Venn Diagrams for Seven Sets

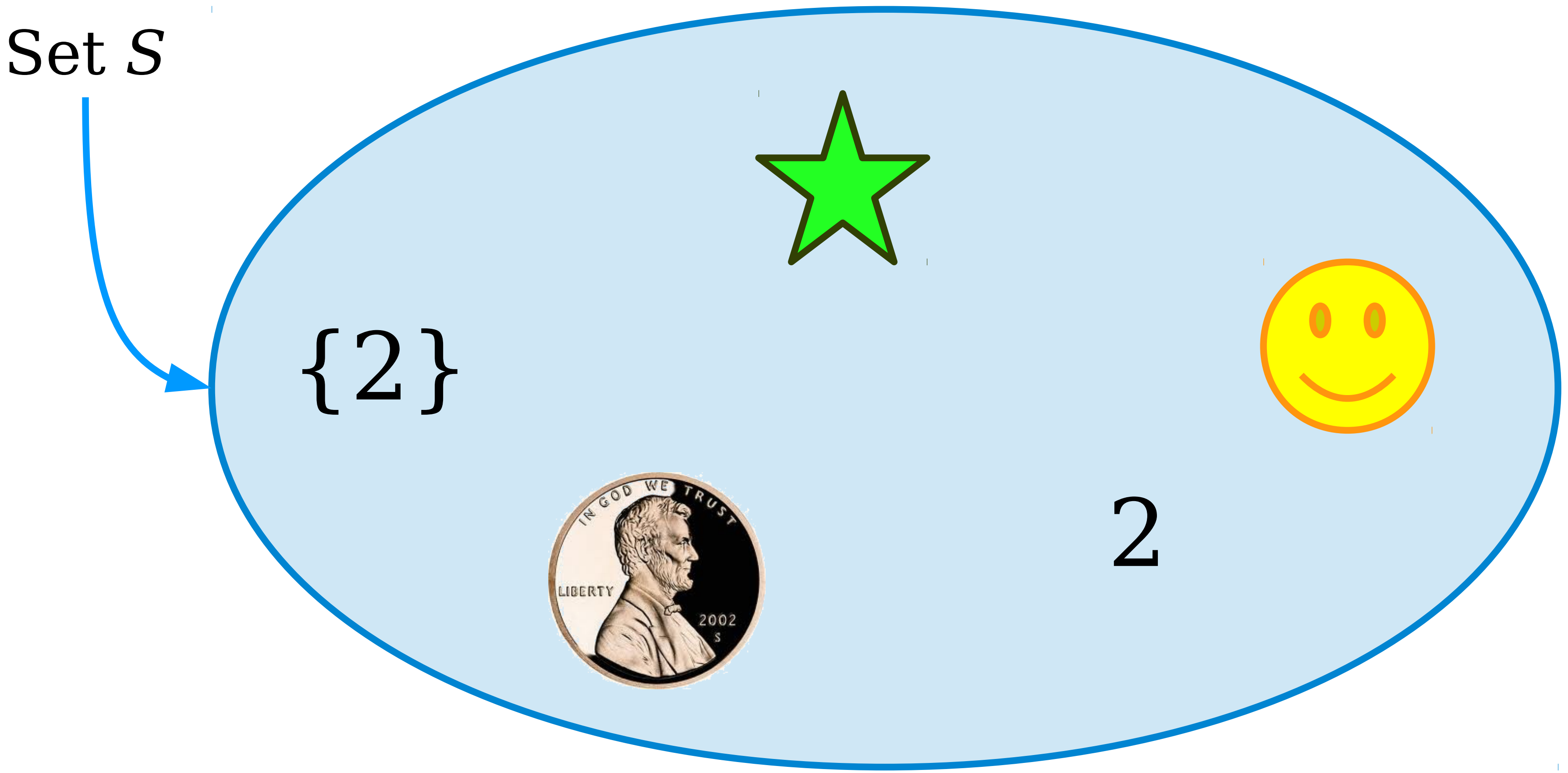
http://moebio.com/research/sevensets/

Subsets and Power Sets

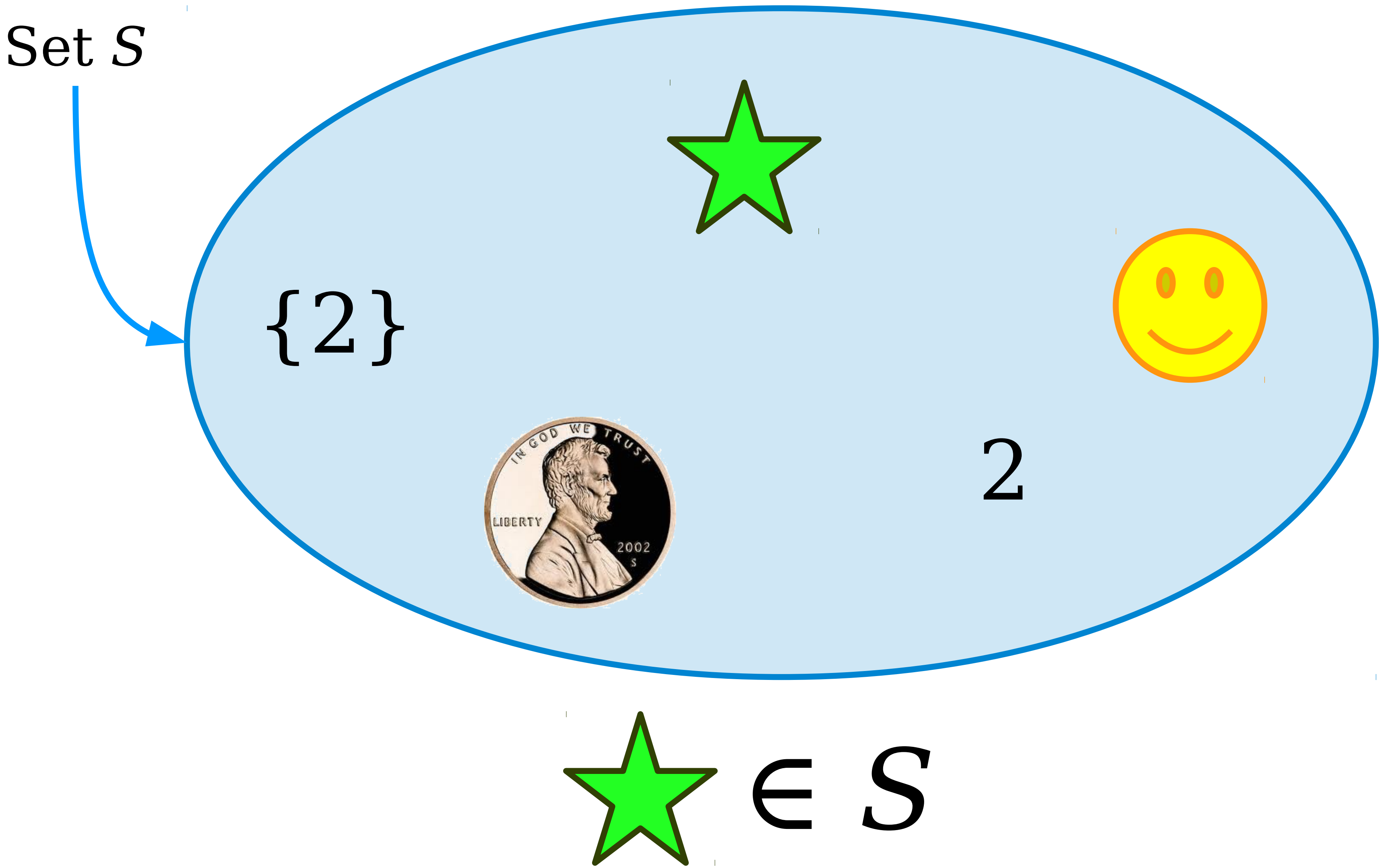
Subsets

- A set S is called a ***subset*** of a set T (denoted $\boldsymbol{S \subseteq T}$) if all elements of S are also elements of T.
- Examples:
  - $\{ 1, 2, 3 \} \subseteq \{ 1, 2, 3, 4 \}$
  - $\{ b, c \} \subseteq \{ a, b, c, d \}$
  - { H, He, Li } ⊆ { H, He, Li }
  - $\mathbb{N} \subseteq \mathbb{Z}$ (*every natural number is an integer*)
  - $\mathbb{Z} \subseteq \mathbb{R}$ (*every integer is a real number*)

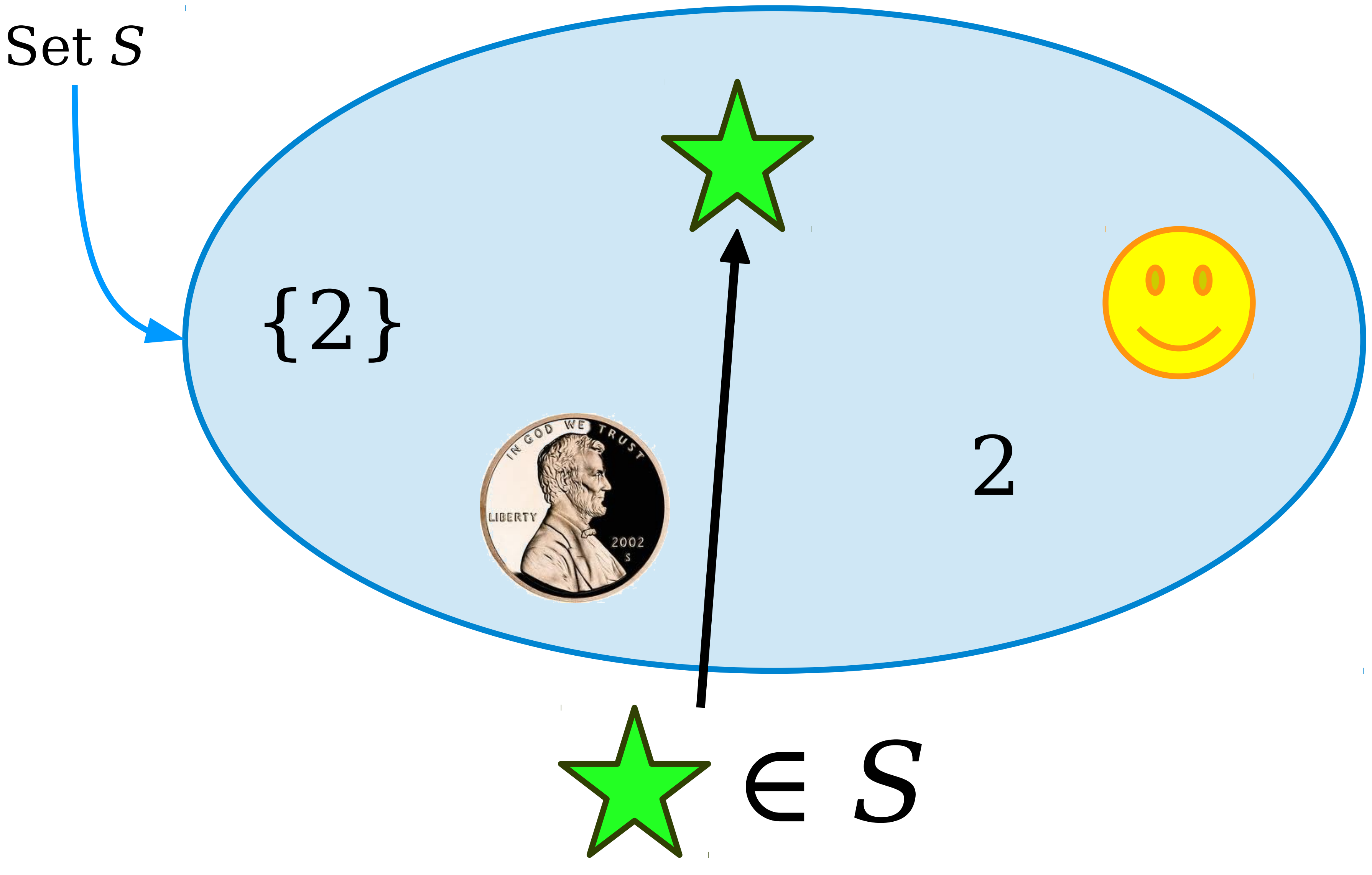
Subsets and Elements



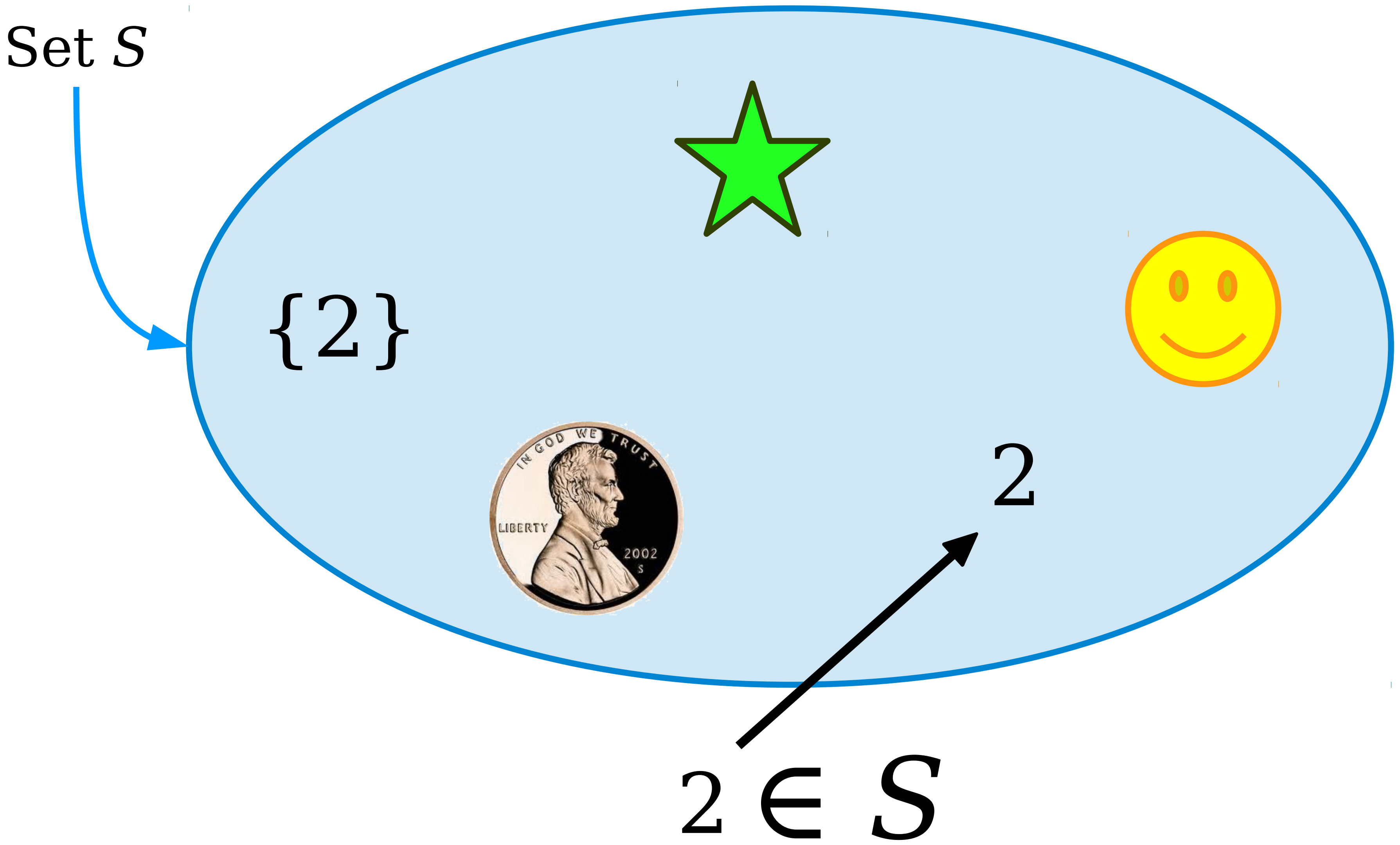
Subsets and Elements



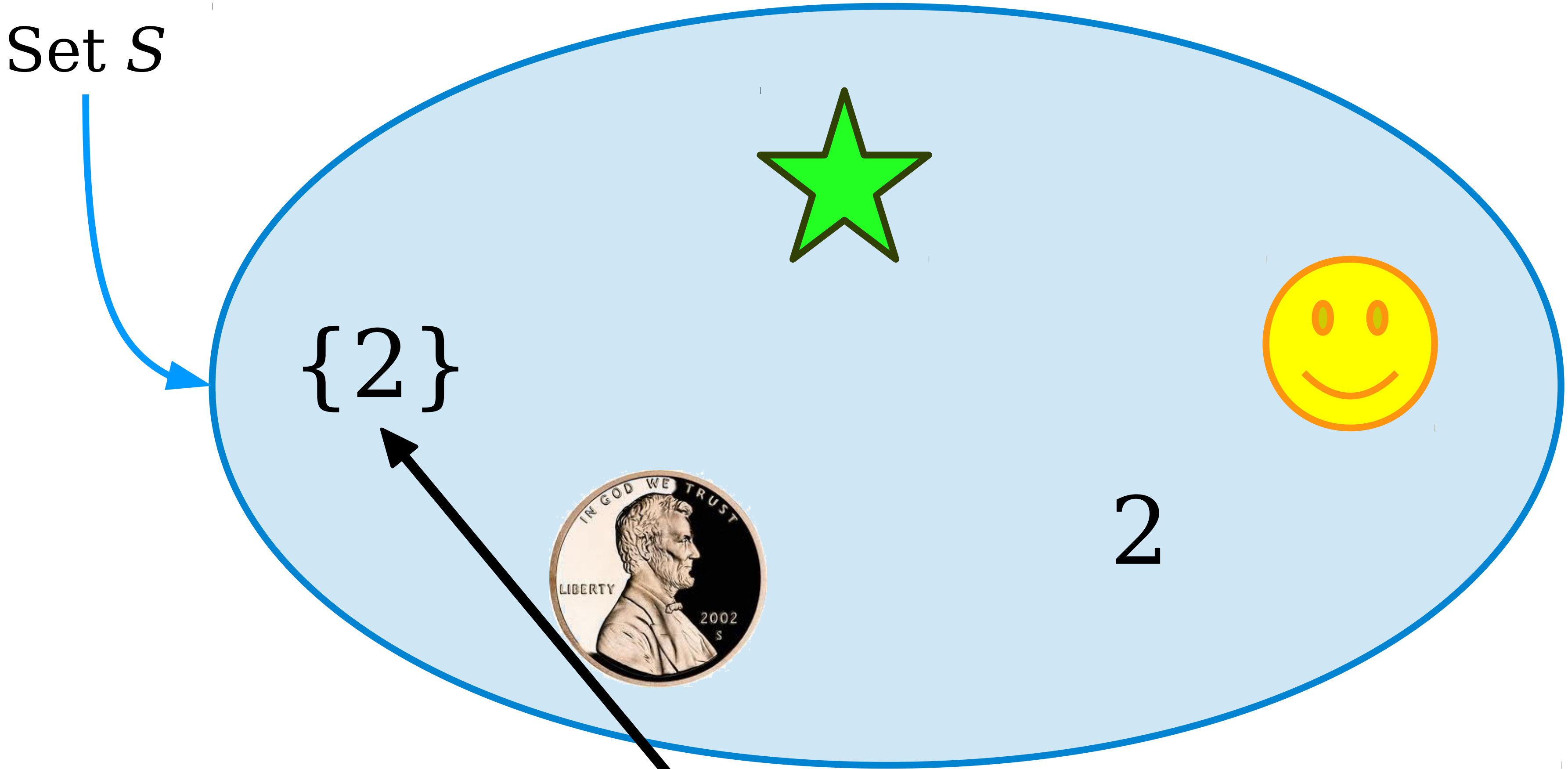
Subsets and Elements



Subsets and Elements

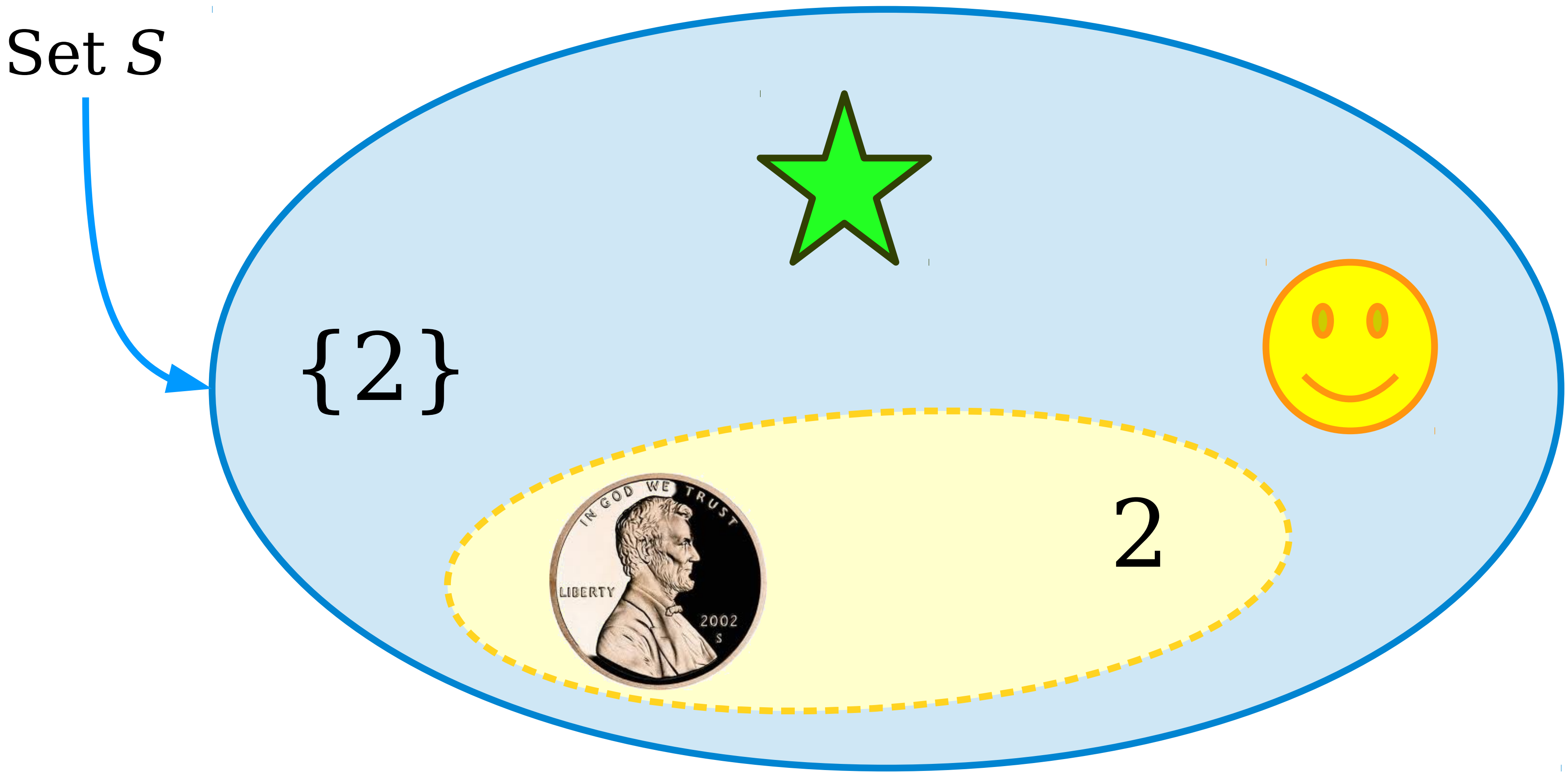


Subsets and Elements

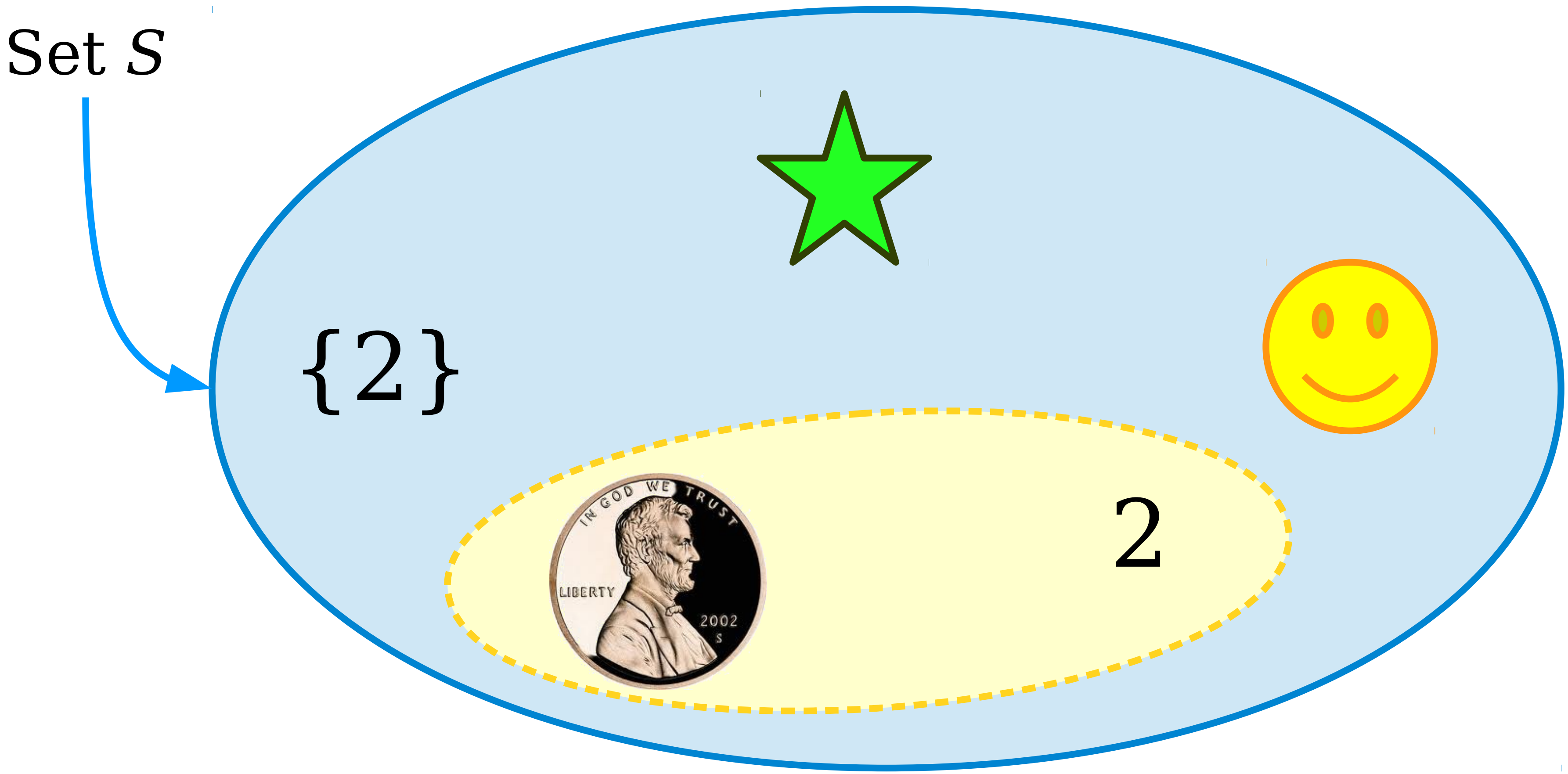


$\{2\} \in S$

Subsets and Elements

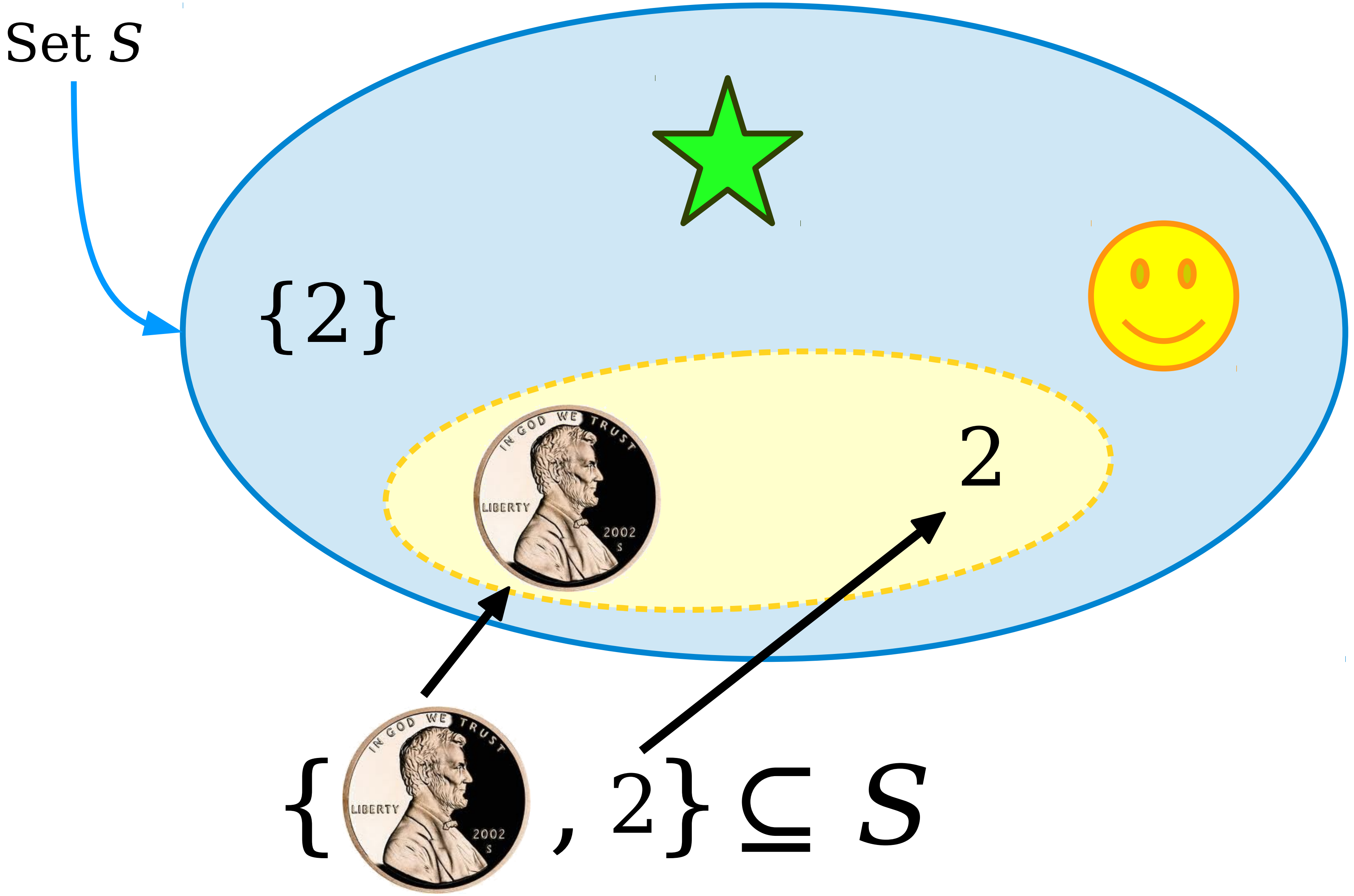


Subsets and Elements

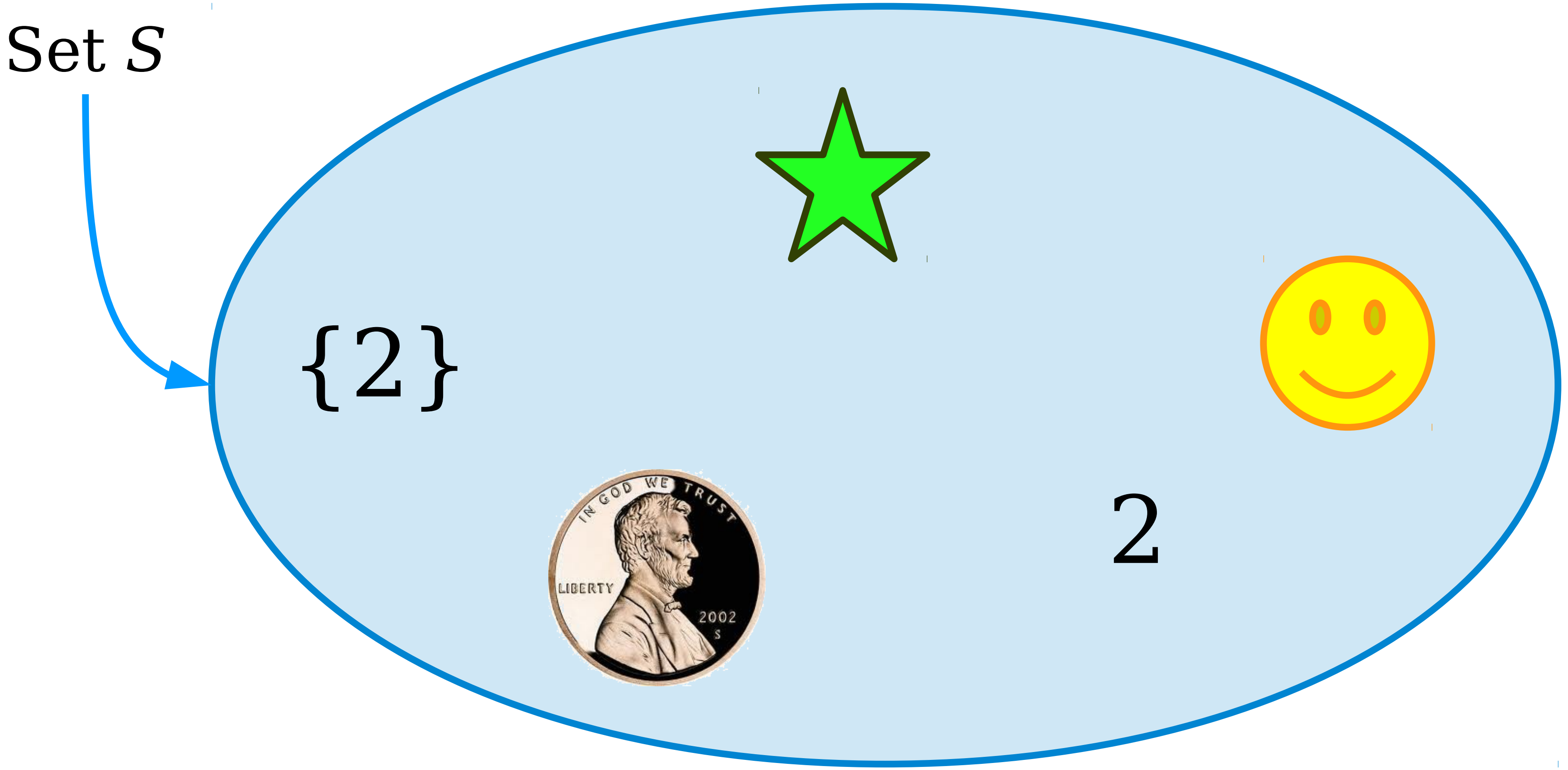


$\{$  $, 2\} \subseteq S$

Subsets and Elements

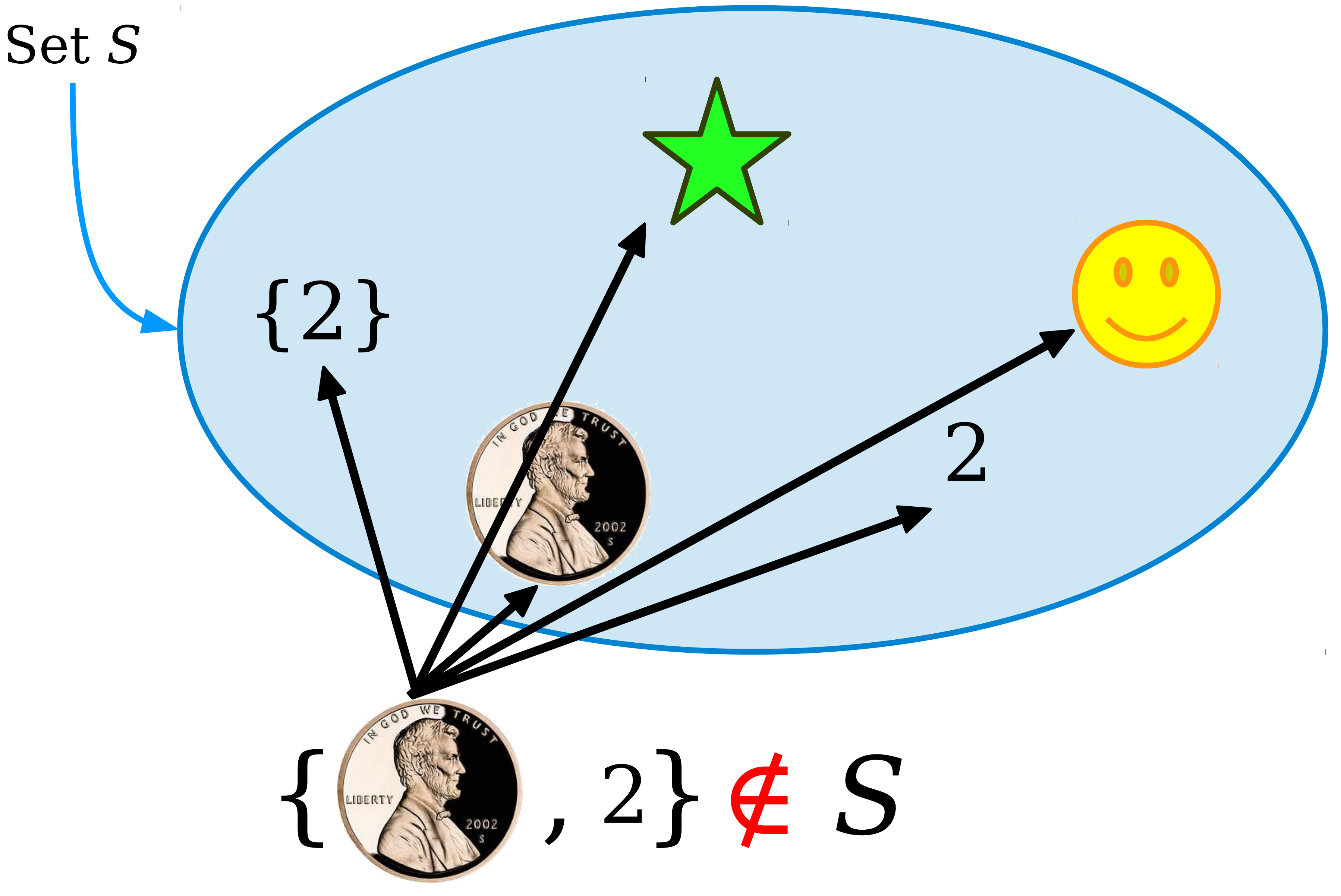


Subsets and Elements

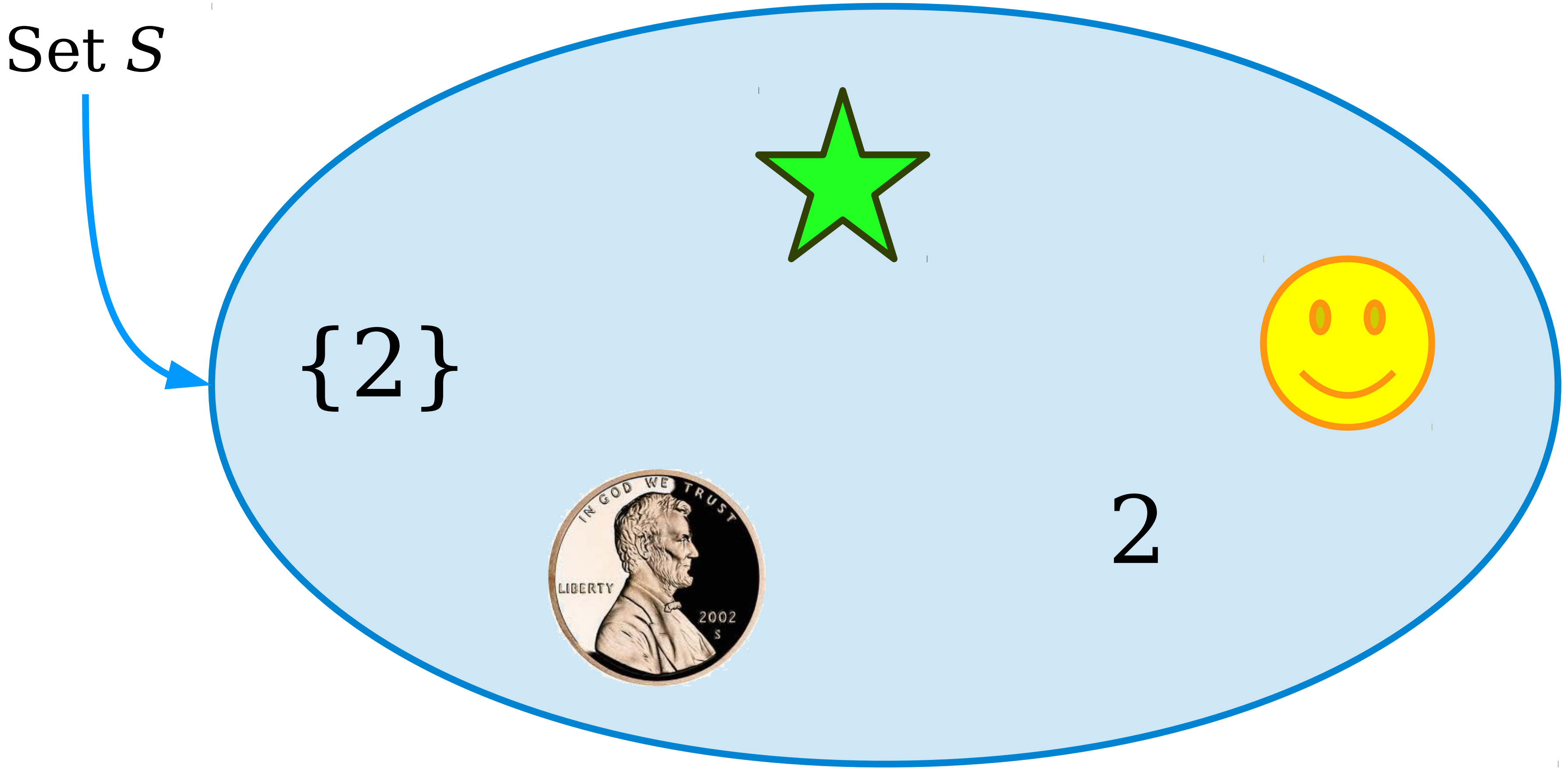


$\{$  $, 2\} \notin S$

Subsets and Elements

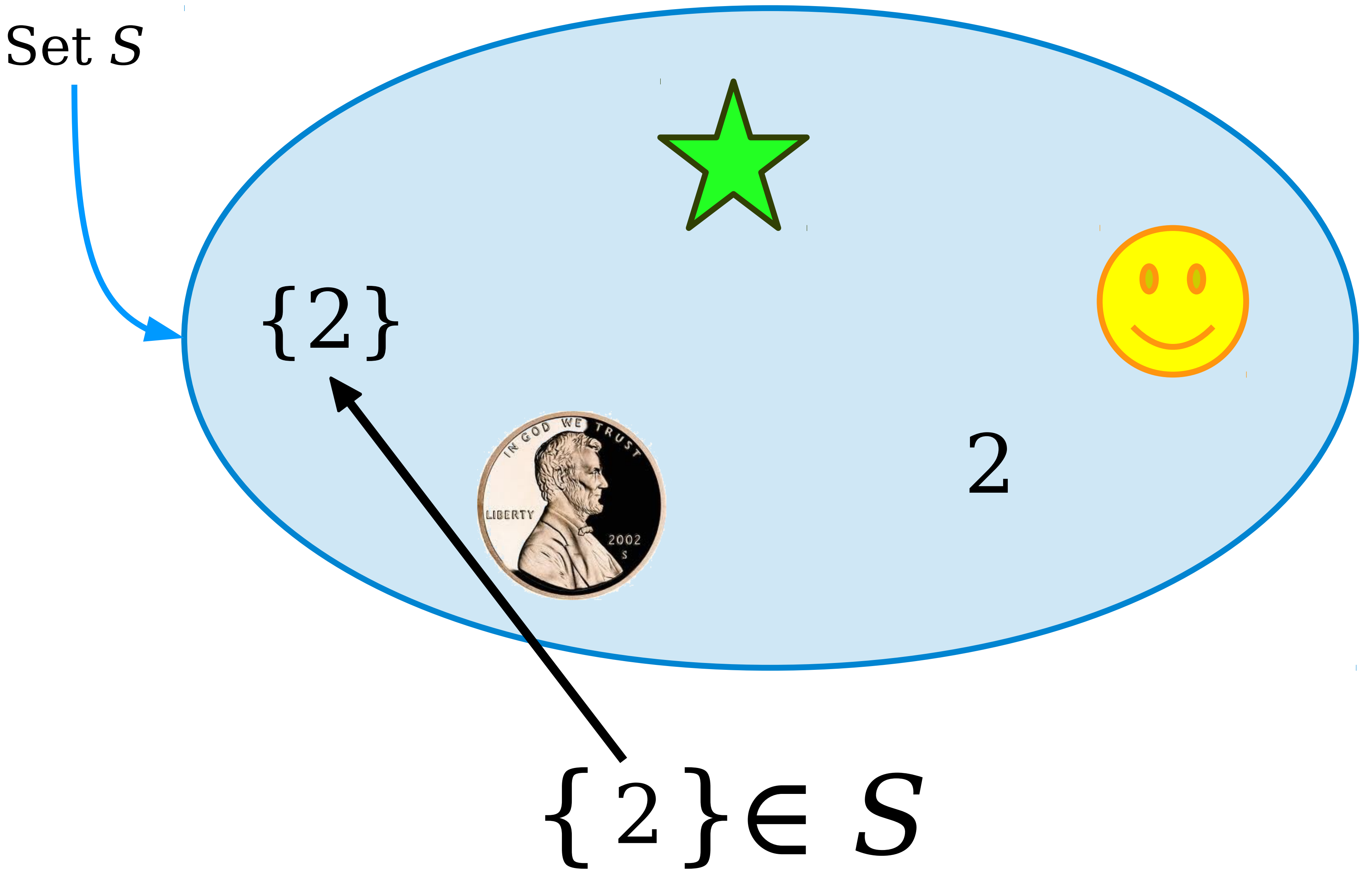


Subsets and Elements

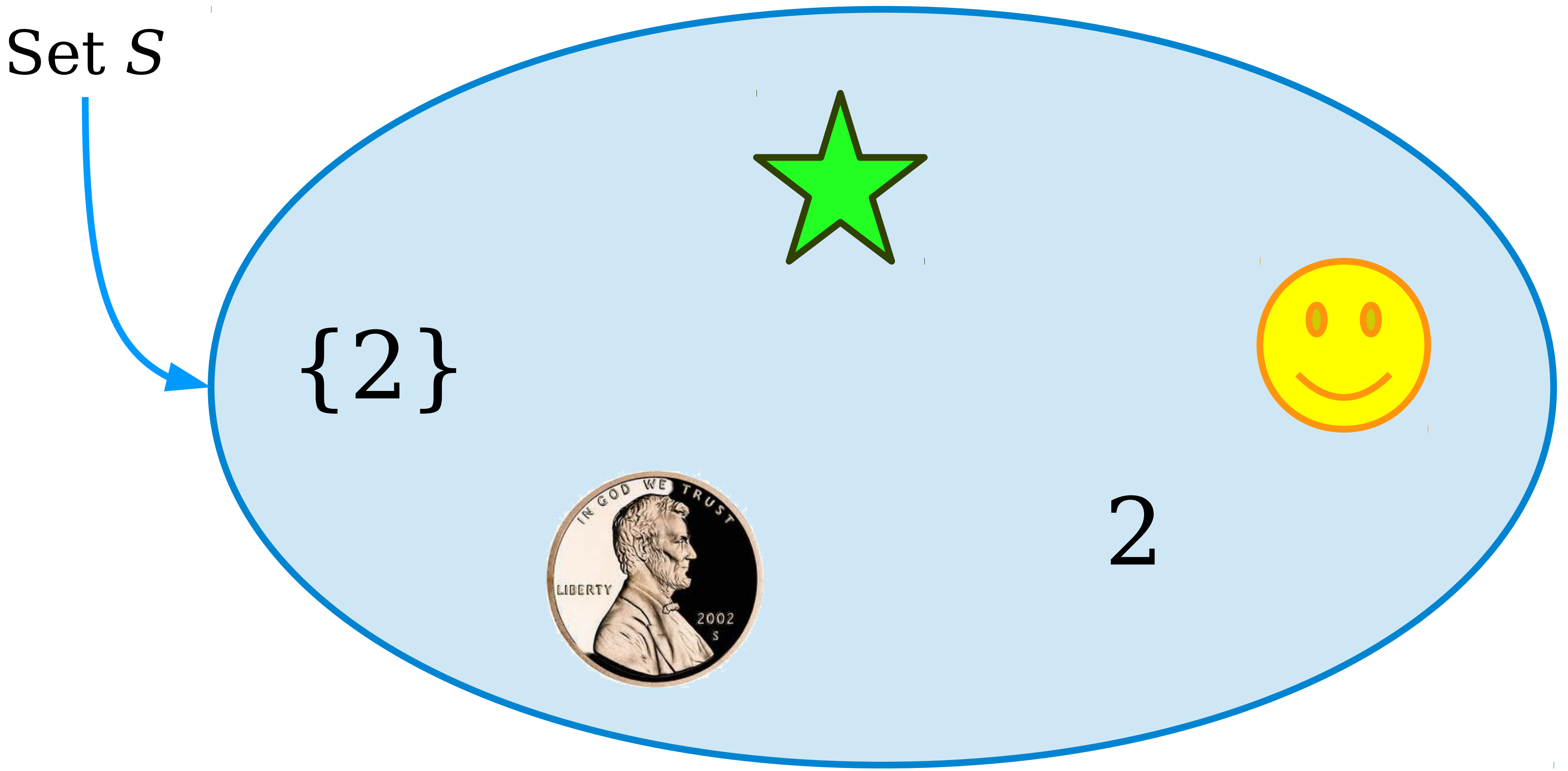


$$\{2\} \in S$$

Subsets and Elements

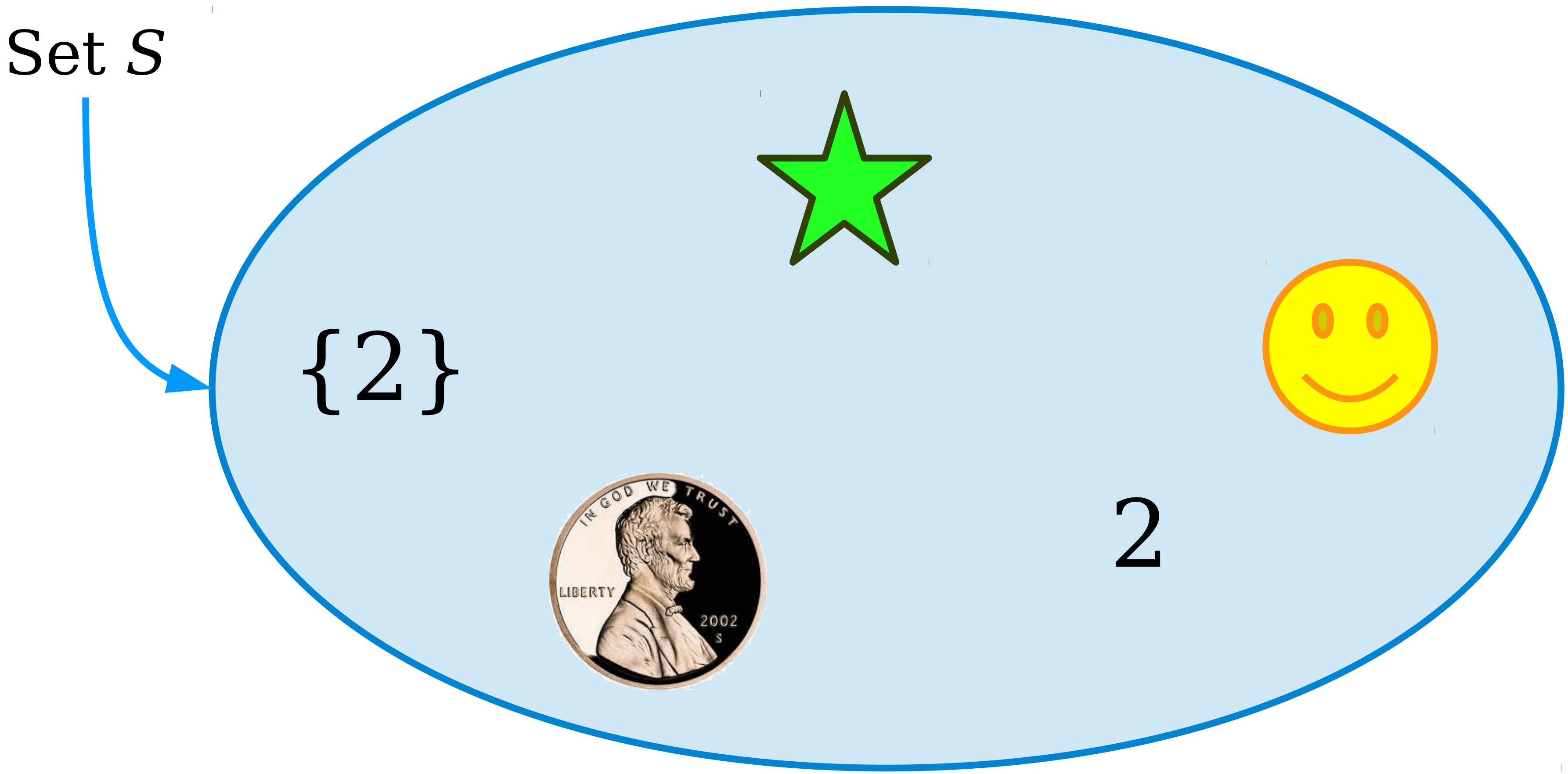


Subsets and Elements



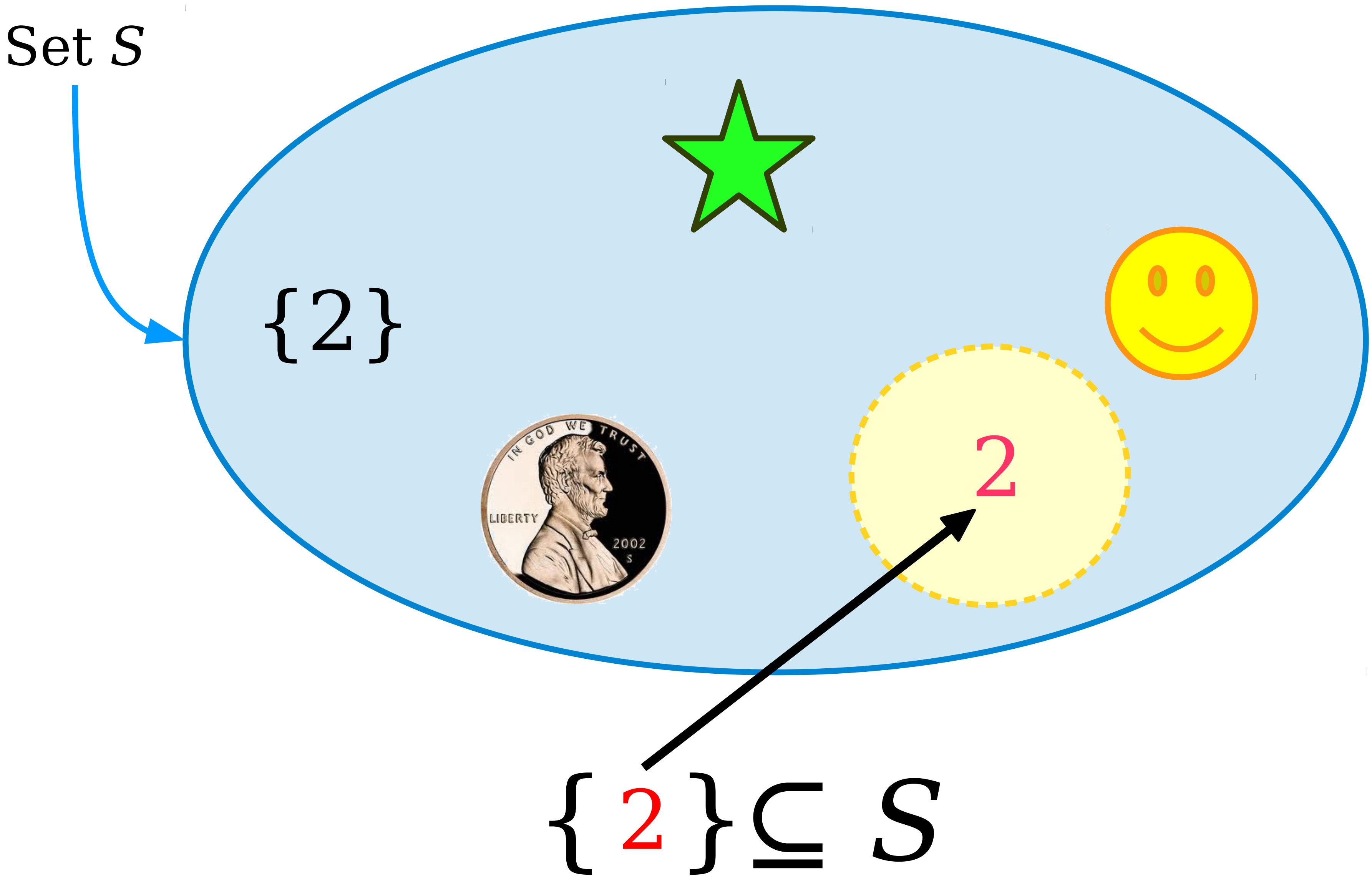
$$\{2\} \subseteq S$$

Subsets and Elements

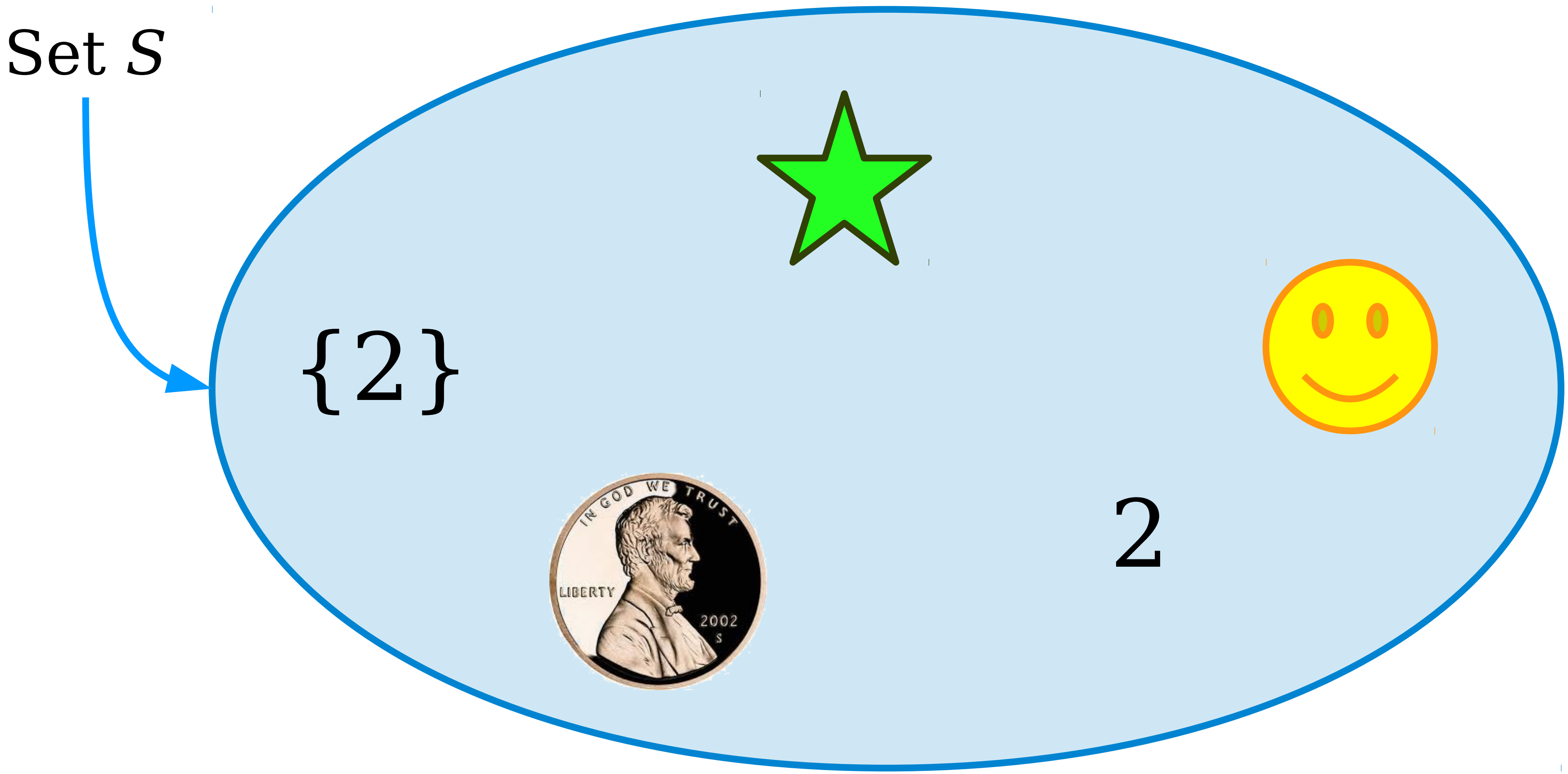


$$\{2\} \subseteq S$$

Subsets and Elements

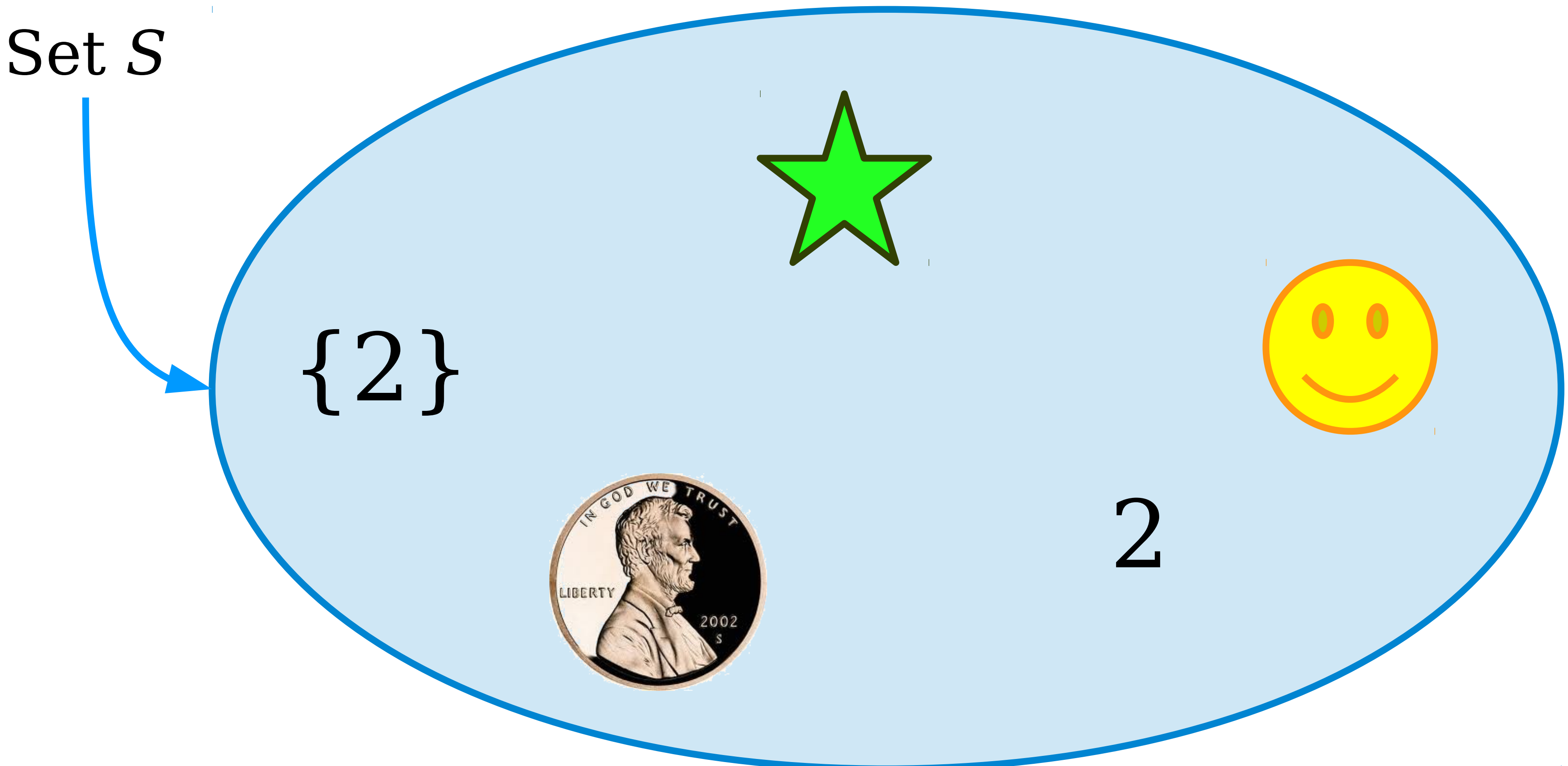


Subsets and Elements



$$2 \not\subseteq S$$

Subsets and Elements



$$2 \not\subseteq S$$

(Since 2 isn't a set.)

Subsets and Elements

- We say that ***$S \in T$*** if, among the elements of T, one of them is *exactly* the object S.
- We say that ***$S \subseteq T$*** if S is a set and every element of S is also an element of T. (S has to be a set for the statement $S \subseteq T$ to be true.)
- Although these concepts are similar, ***they are not the same!*** Not all elements of a set are subsets of that set and vice-versa.
- We have a resource on the course website, the Guide to Elements and Subsets, that explores this in more depth.

What About the Empty Set?

- A set S is called a ***subset*** of a set T (denoted $\boldsymbol{S \subseteq T}$) if all elements of S are also elements of T.
- Are there any sets T where $\emptyset \subseteq T$?
- Equivalently, is there a set T where the following statement is true?

"All elements of Ø are also elements of *T*"

- ***Yes!*** In fact, this statement is true for *every* set T!

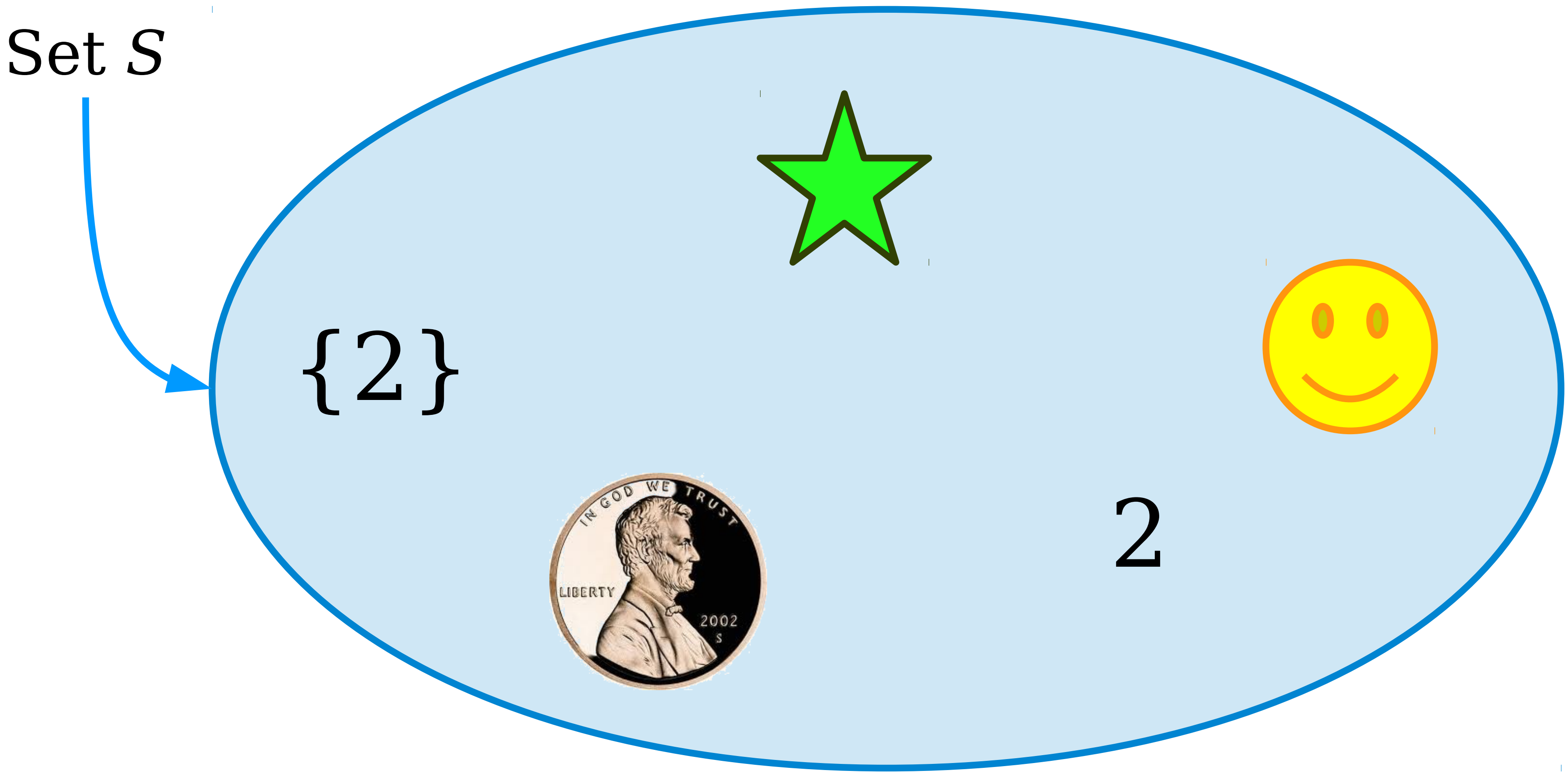
Vacuous Truth

- A statement of the form

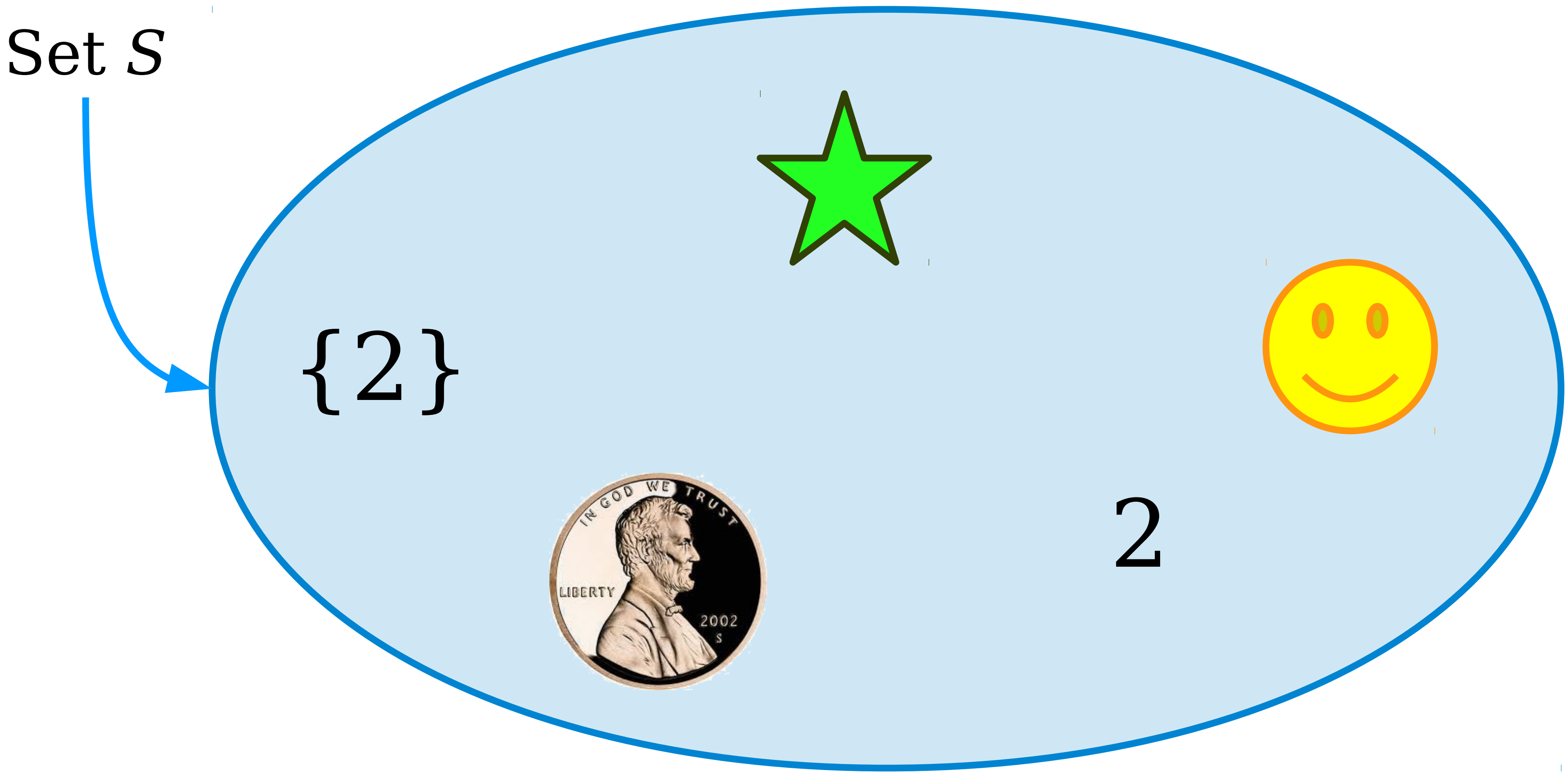
  "All objects of type *P*
  are also of type *Q*"

  is called ***vacuously true*** if there are no objects of type P.
- Vacuously true statements are true *by definition*. This is a convention used throughout mathematics.
- Some examples:
  - All unicorns are pink.
  - All unicorns are blue.
  - Every element of Ø is also an element of T.

Subsets and Elements

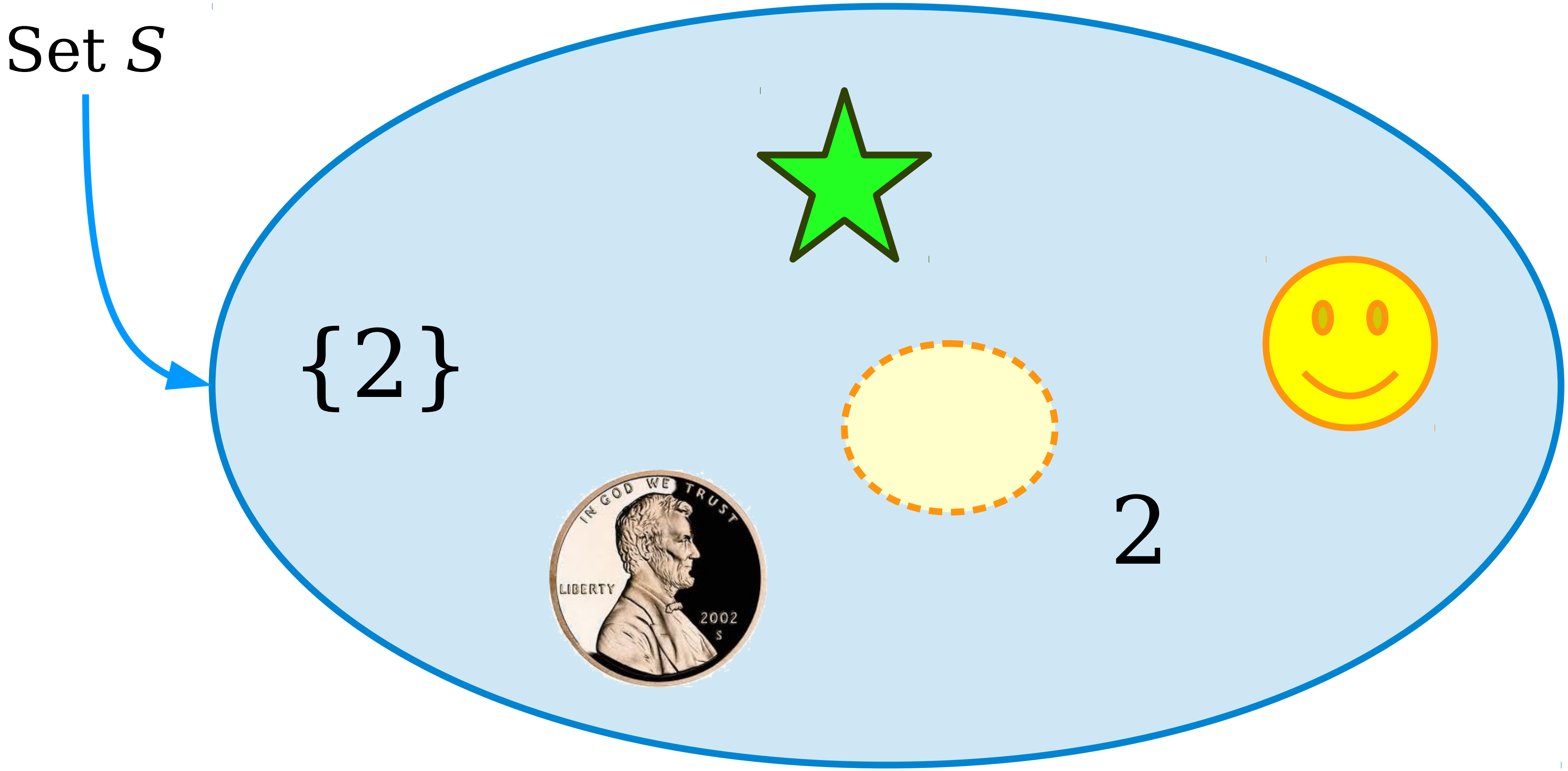


Subsets and Elements



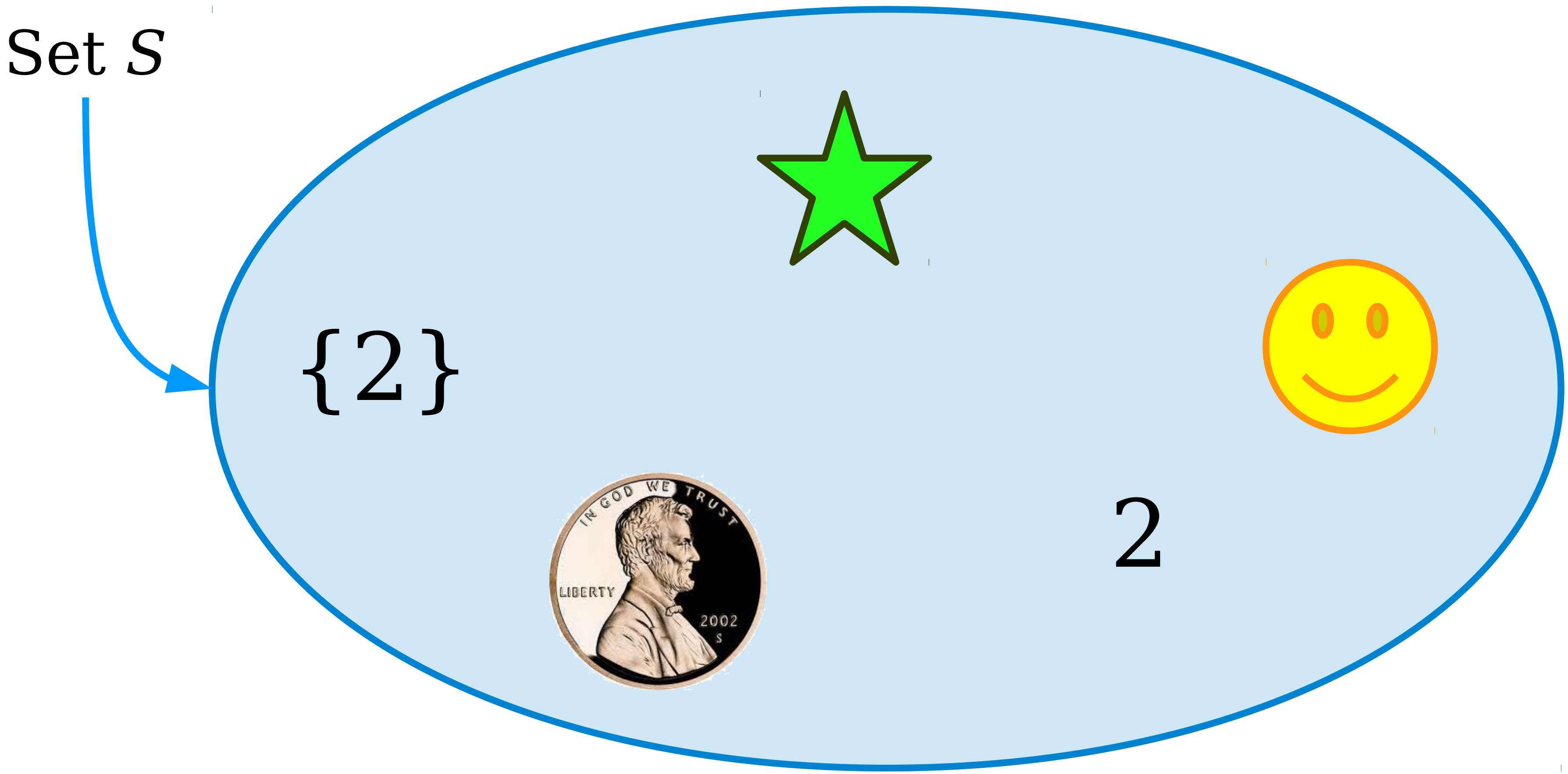
$$\emptyset \subseteq S$$

Subsets and Elements



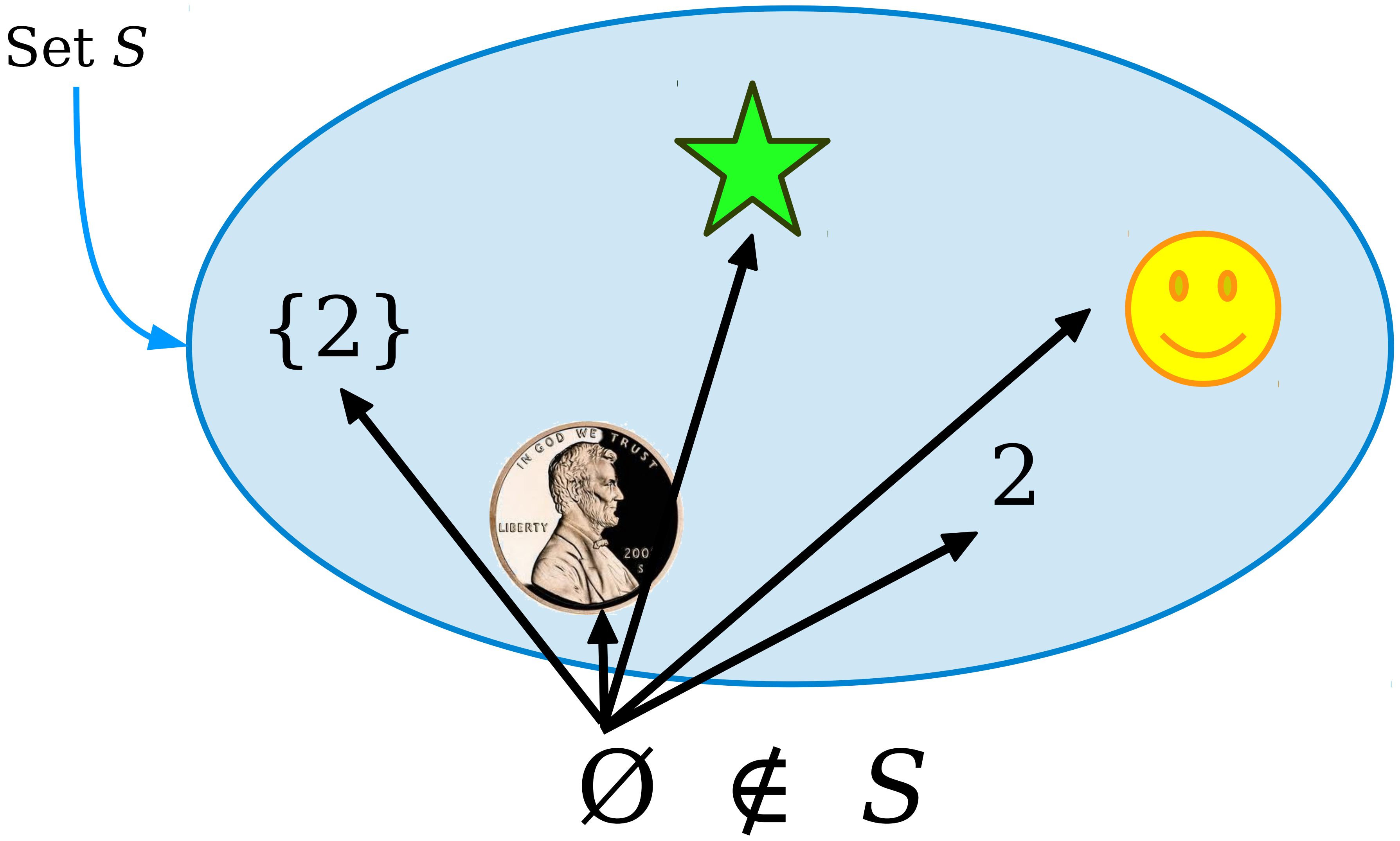
$$\emptyset \subseteq S$$

Subsets and Elements



$$\emptyset \notin S$$

Subsets and Elements



$S = \{$  ,  $\}$

$\wp(S) = \{\ \emptyset,\ \{$  $\},\ \{$  $\},\ \{$  ,  $\}\ \}$

This is the ***power set*** of S, the set of all subsets of S. We write the power set of S as $\wp(S)$.

Formally, $\wp(S) = \{\ T \mid T \subseteq S\ \}$.

(Do you see why?)

What is $\wp(\emptyset)$?

Answer: $\{\emptyset\}$

Remember that $\emptyset \neq \{\emptyset\}$!

Let’s take a quick 5 minute break!

Cardinality

Cardinality

- The ***cardinality*** of a set is the number of elements it contains.
- If S is a set, we denote its cardinality by writing $|S|$.
- Examples:
  - $|\{\mathbf{38}, \mathbf{31}\}| = 2$
  - $|\{\{a, b\}, \{c, d, e, f, g\}, \{h\}\}| = 3$
  - $|\{1, 2, 3, 3, 3, 3, 3\}| = 3$
  - $|\{ n \in \mathbb{N} \mid n < 137 \}| = 137$

The Cardinality of $\mathbb{N}$

- What is $|\mathbb{N}|$?
  - There are infinitely many natural numbers.
  - $|\mathbb{N}|$ can't be a natural number, since it's infinitely large.

The Cardinality of $\mathbb{N}$

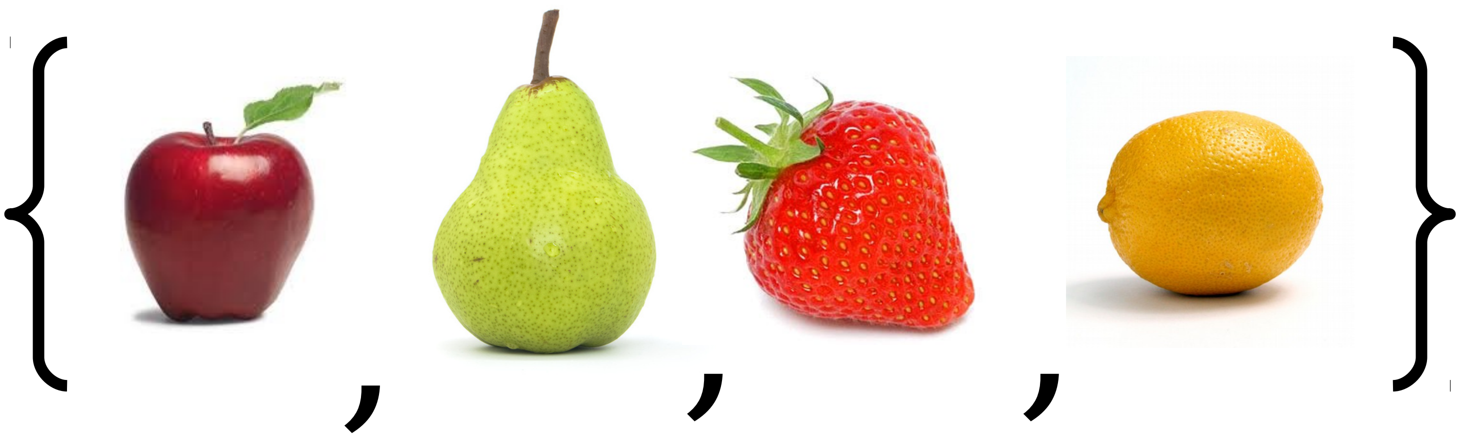
- What is $|\mathbb{N}|$?
  - There are infinitely many natural numbers.
  - $|\mathbb{N}|$ can't be a natural number, since it's infinitely large.
- We need to introduce a new term.
- Let's define $\aleph_0 = |\mathbb{N}|$.
  - $\aleph_0$ is pronounced “aleph-zero,” “aleph-nought,” or “aleph-null.”

Consider the set

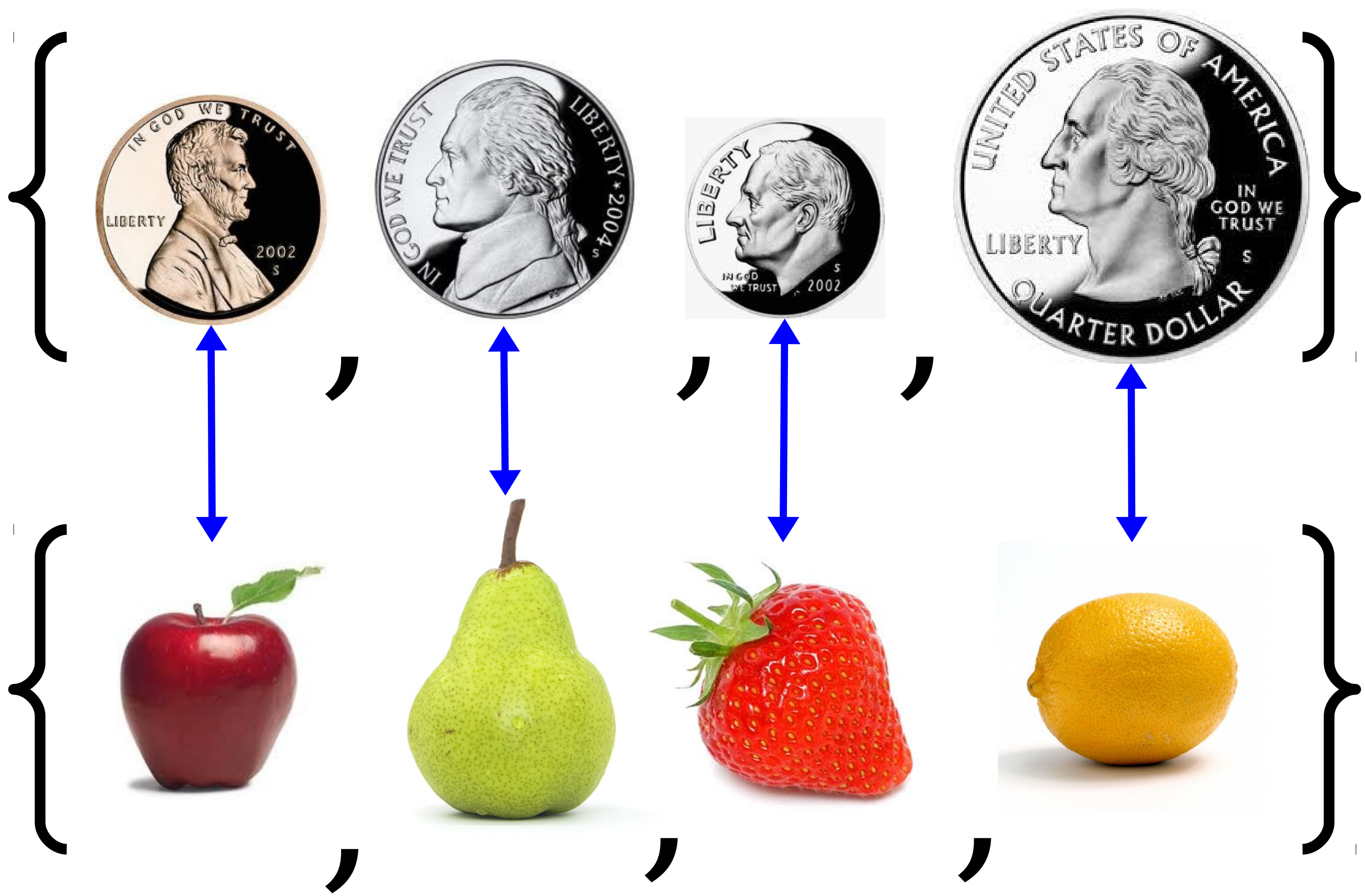
$$S = \{\ n \mid n \in \mathbb{N} \text{ and } n \text{ is even }\}$$

What is $|S|$?

How Big Are These Sets?

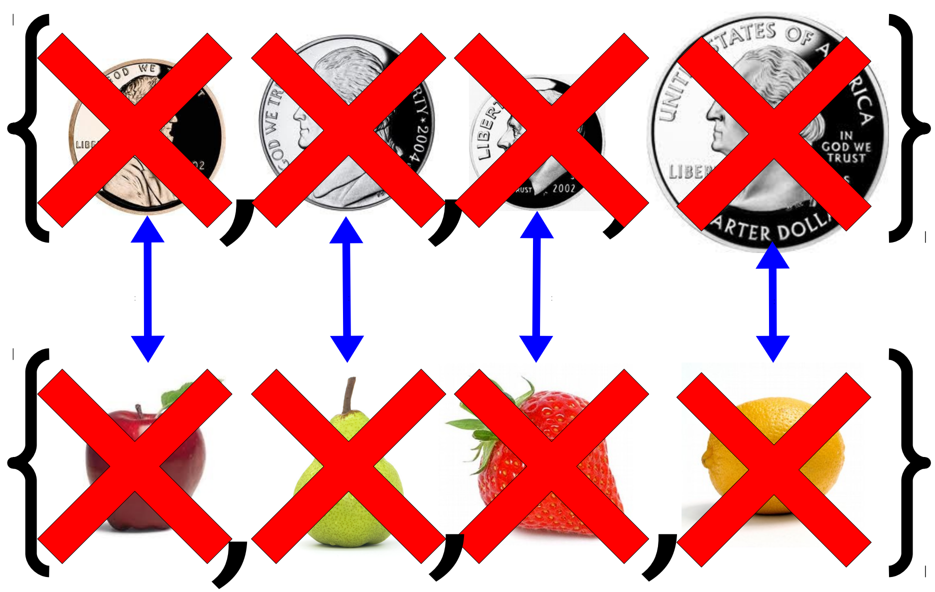


How Big Are These Sets?



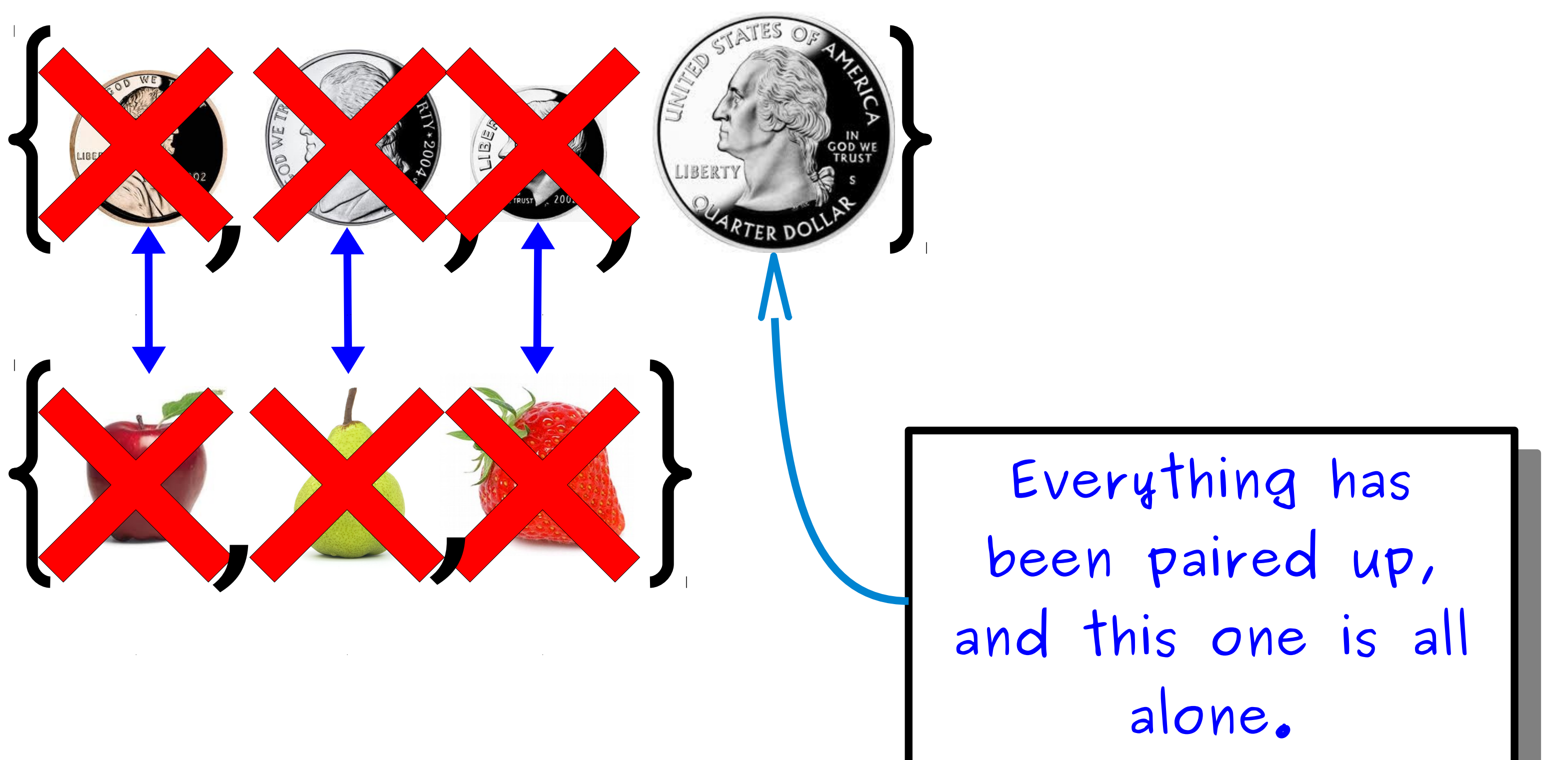
Comparing Cardinalities

- *By definition*, two sets have the same size if there is a way to pair their elements off without leaving any elements uncovered.
- The intuition:

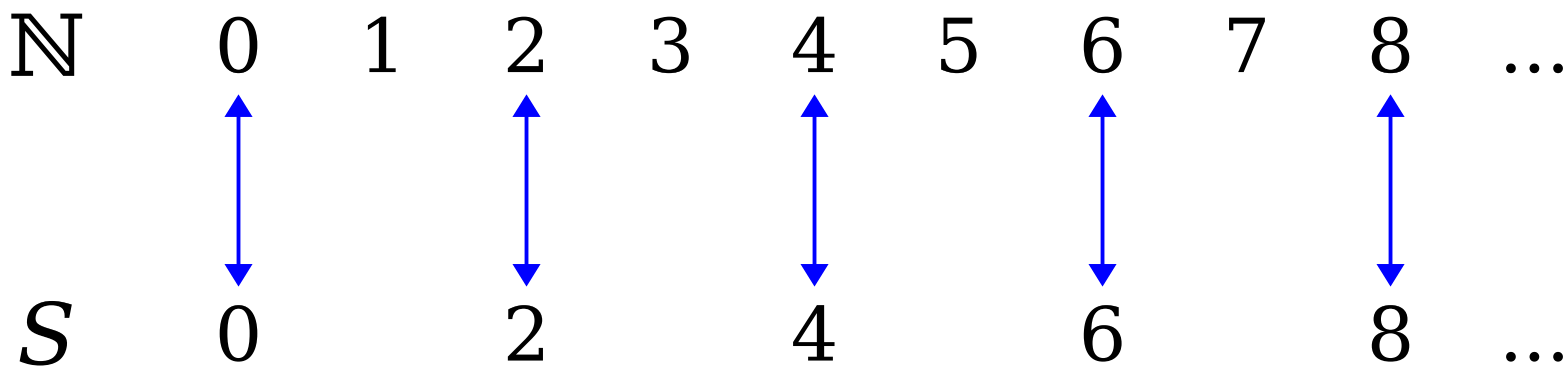


Comparing Cardinalities

- *By definition*, two sets have the same size if there is a way to pair their elements off without leaving any elements uncovered.
- The intuition:



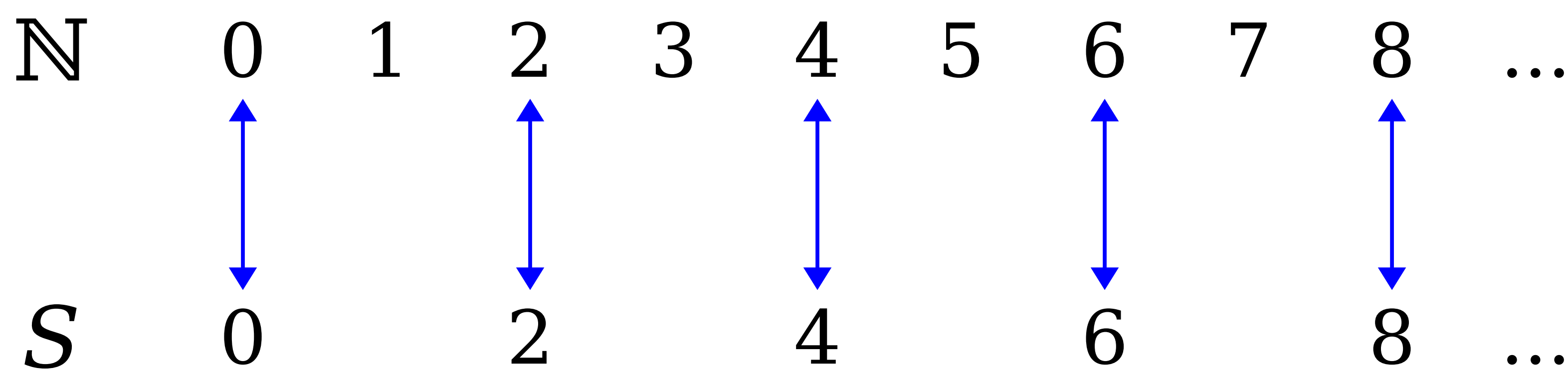
Infinite Cardinalities



$$S = \{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even } \}$$

Two sets have the same size if there is a way to pair their elements off without leaving any elements uncovered

Infinite Cardinalities



$$S = \{\ n \mid n \in \mathbb{N} \text{ and } n \text{ is even }\}$$

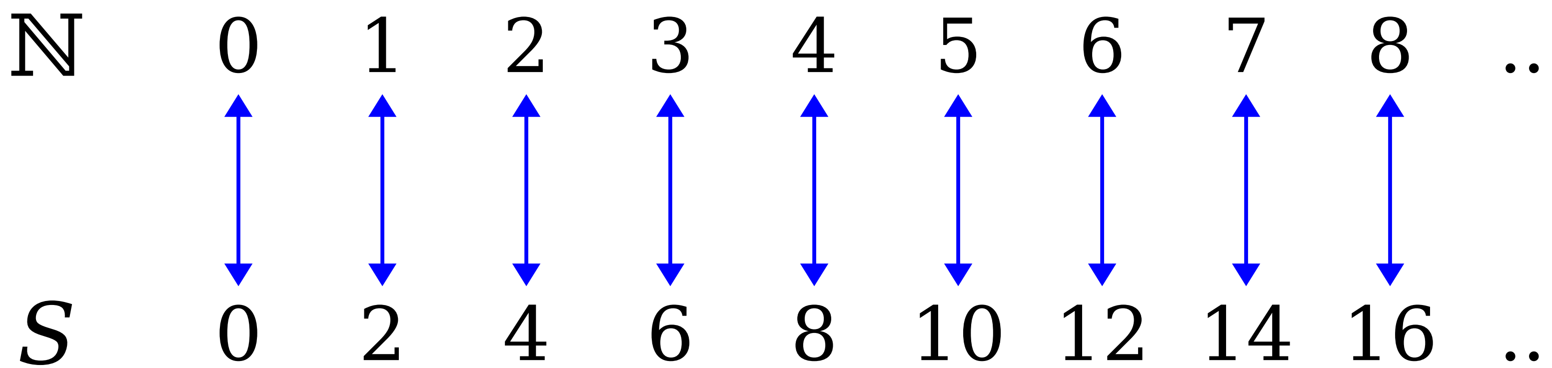
Two sets have the same size if *there is a way* to pair their elements off without leaving any elements uncovered

Infinite Cardinalities

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	…
S	0		2		4		6		8	…

$$S = \{\ n \mid n \in \mathbb{N} \text{ and } n \text{ is even }\}$$

Infinite Cardinalities



$$n \leftrightarrow 2n$$

$$S = \{\ n \mid n \in \mathbb{N} \text{ and } n \text{ is even }\}$$

$$|S| = |\mathbb{N}| = \aleph_0$$

Infinite Cardinalities

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	…
$\mathbb{Z}$	…	-3	-2	-1	0	1	2	3	4	…

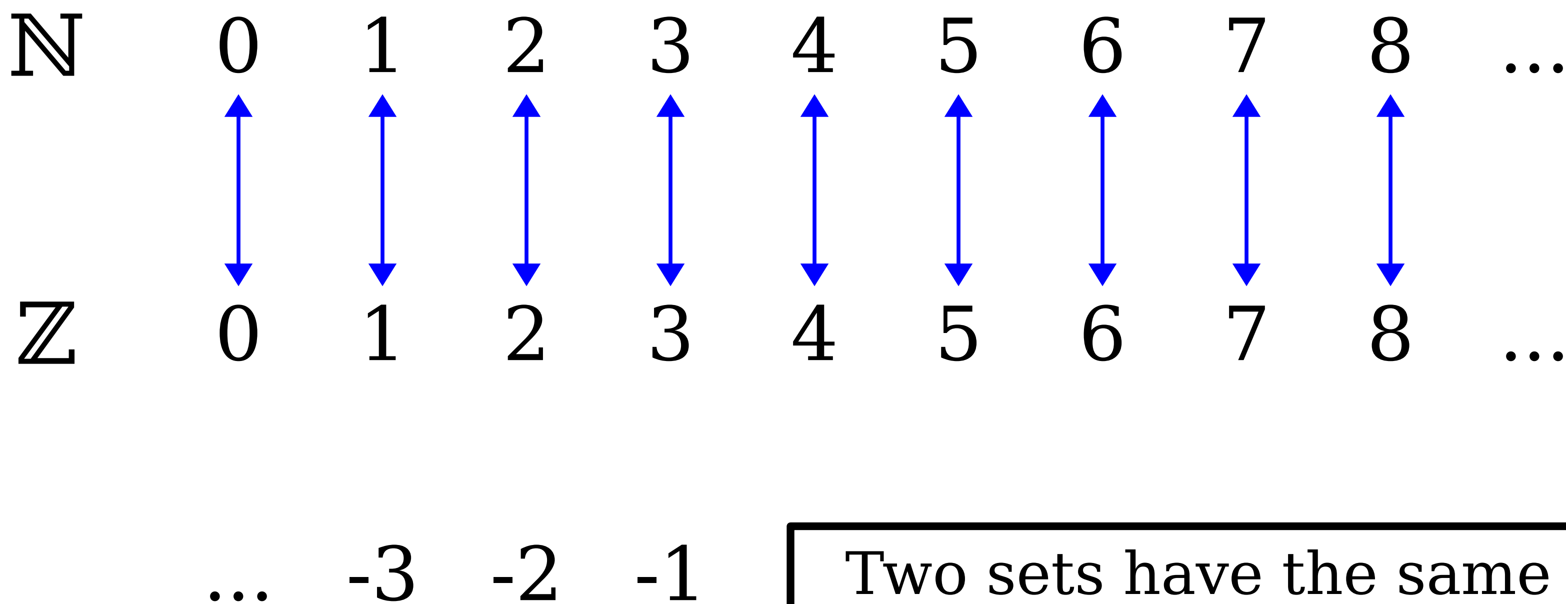
Infinite Cardinalities

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	…
$\mathbb{Z}$					0	1	2	3	4	…
	…	-3	-2	-1						

Infinite Cardinalities

ℕ	0	1	2	3	4	5	6	7	8	...
	↕	↕	↕	↕	↕	↕	↕	↕	↕	
ℤ	0	1	2	3	4	5	6	7	8	...
	...	-3	-2	-1						

Infinite Cardinalities



Two sets have the same size if *there is a way* to pair their elements off without leaving any elements uncovered

Infinite Cardinalities

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	…
$\mathbb{Z}$	…	-3	-2	-1	0	1	2	3	4	…

Infinite Cardinalities

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	...

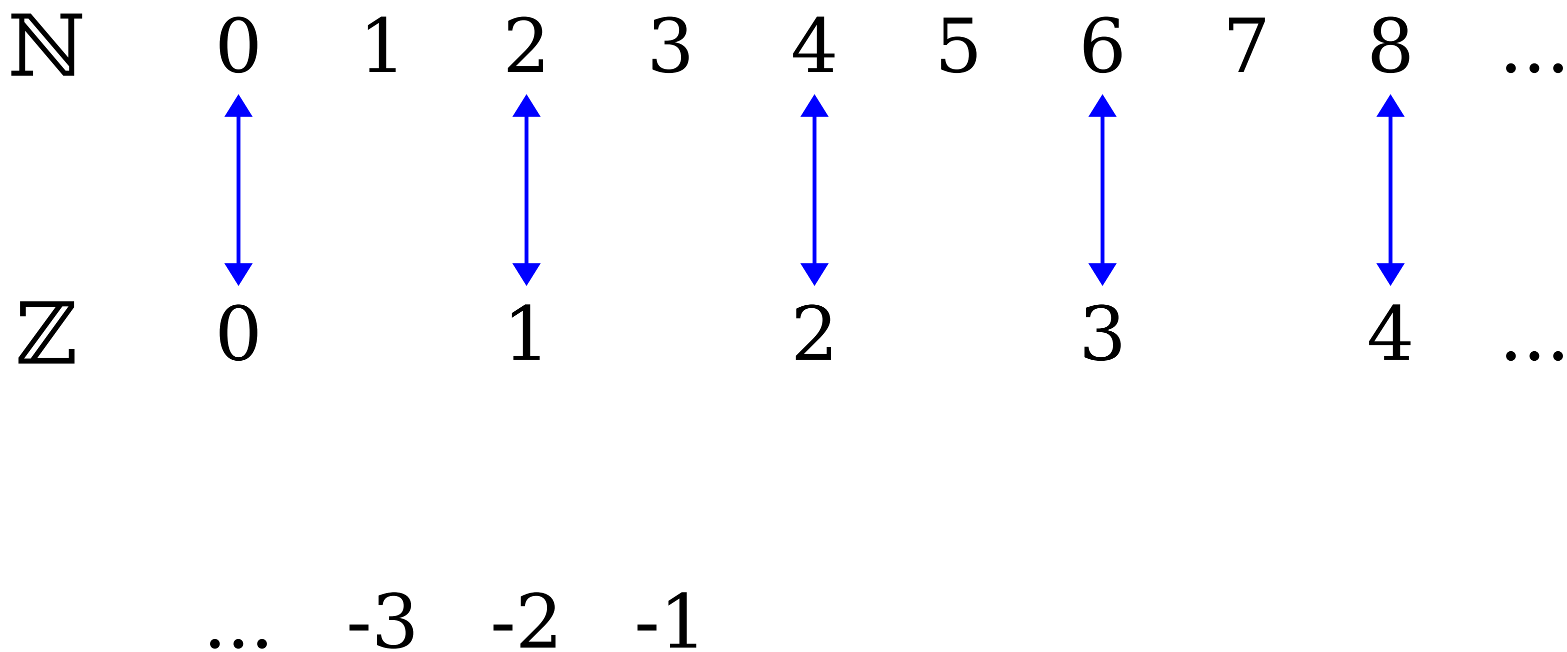
$\mathbb{Z}$

...	-3	-2	-1	0	1	2	3	4	...

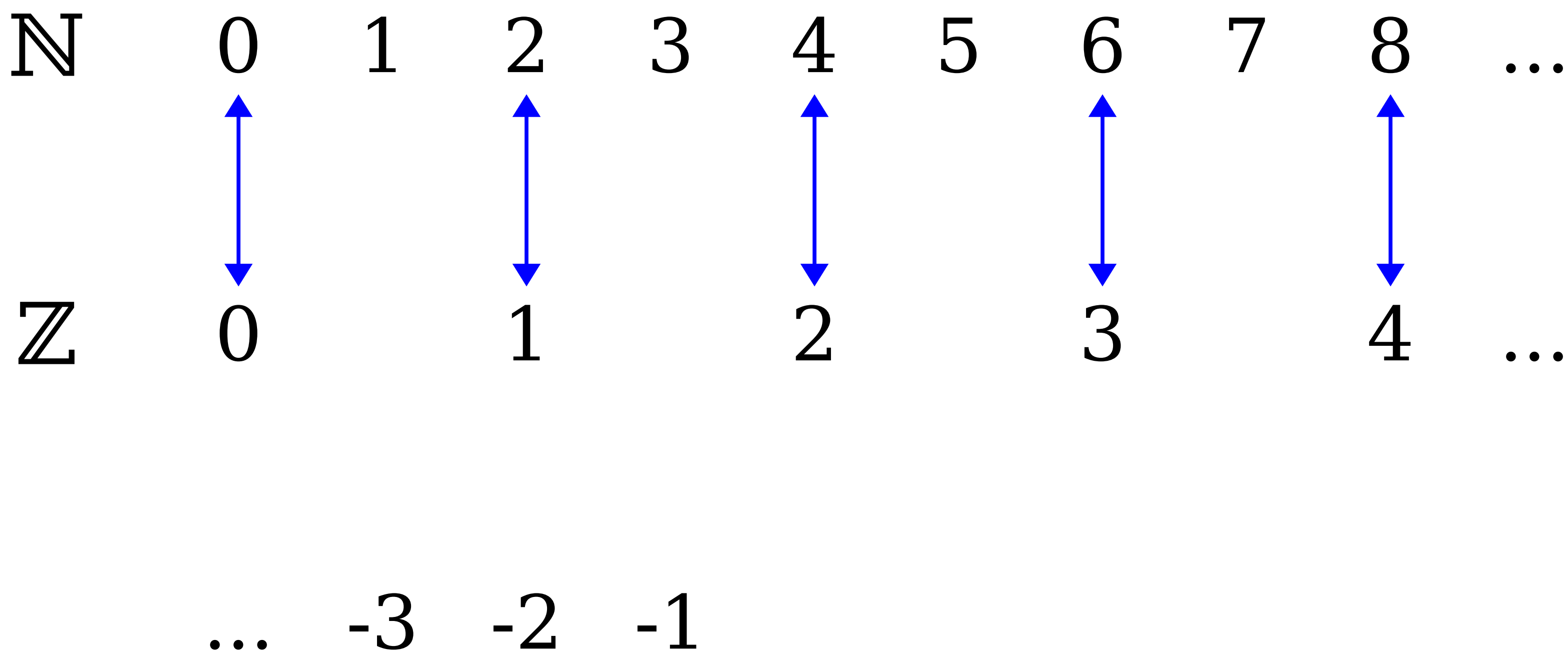
Infinite Cardinalities

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	...
$\mathbb{Z}$	0		1		2		3		4	...
	...	-3	-2	-1						

Infinite Cardinalities

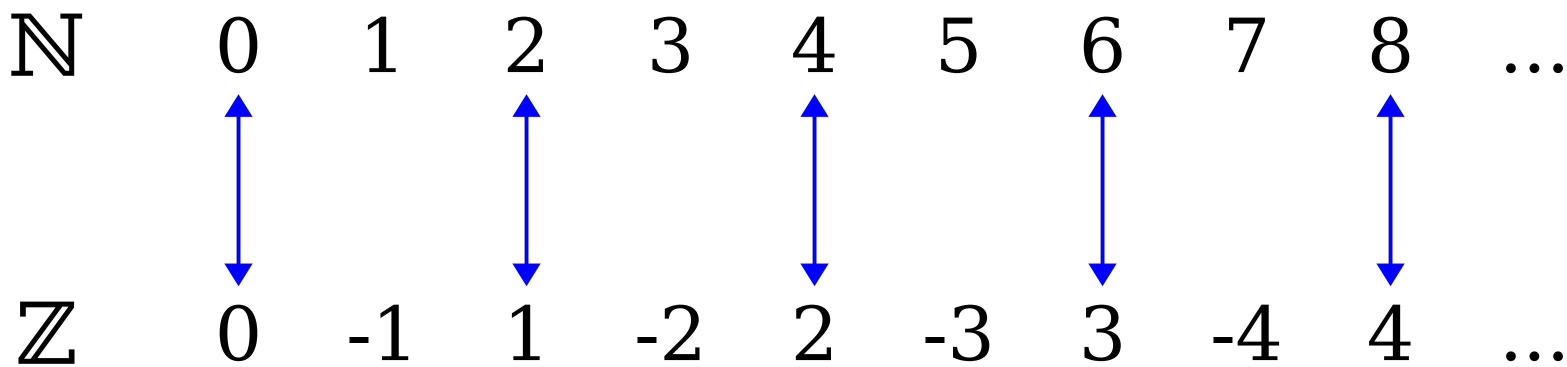


Infinite Cardinalities



Pair nonnegative integers with even natural numbers.

Infinite Cardinalities



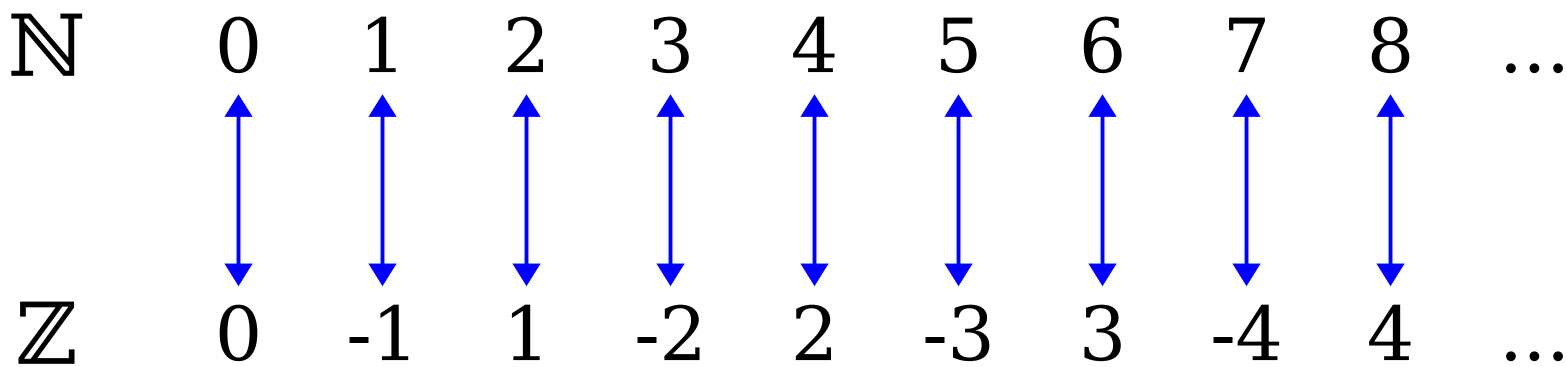
Pair nonnegative integers with even natural numbers.

Infinite Cardinalities

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	...
$\mathbb{Z}$	0	-1	1	-2	2	-3	3	-4	4	...

Pair nonnegative integers with even natural numbers.

Infinite Cardinalities



Pair nonnegative integers with even natural numbers.
Pair negative integers with odd natural numbers.

Infinite Cardinalities

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	...
$\mathbb{Z}$	0	-1	1	-2	2	-3	3	-4	4	...

$$|\mathbb{N}| = |\mathbb{Z}| = \aleph_0$$

Pair nonnegative integers with even natural numbers.
Pair negative integers with odd natural numbers.

Important Question:

Do all infinite sets have the same cardinality?

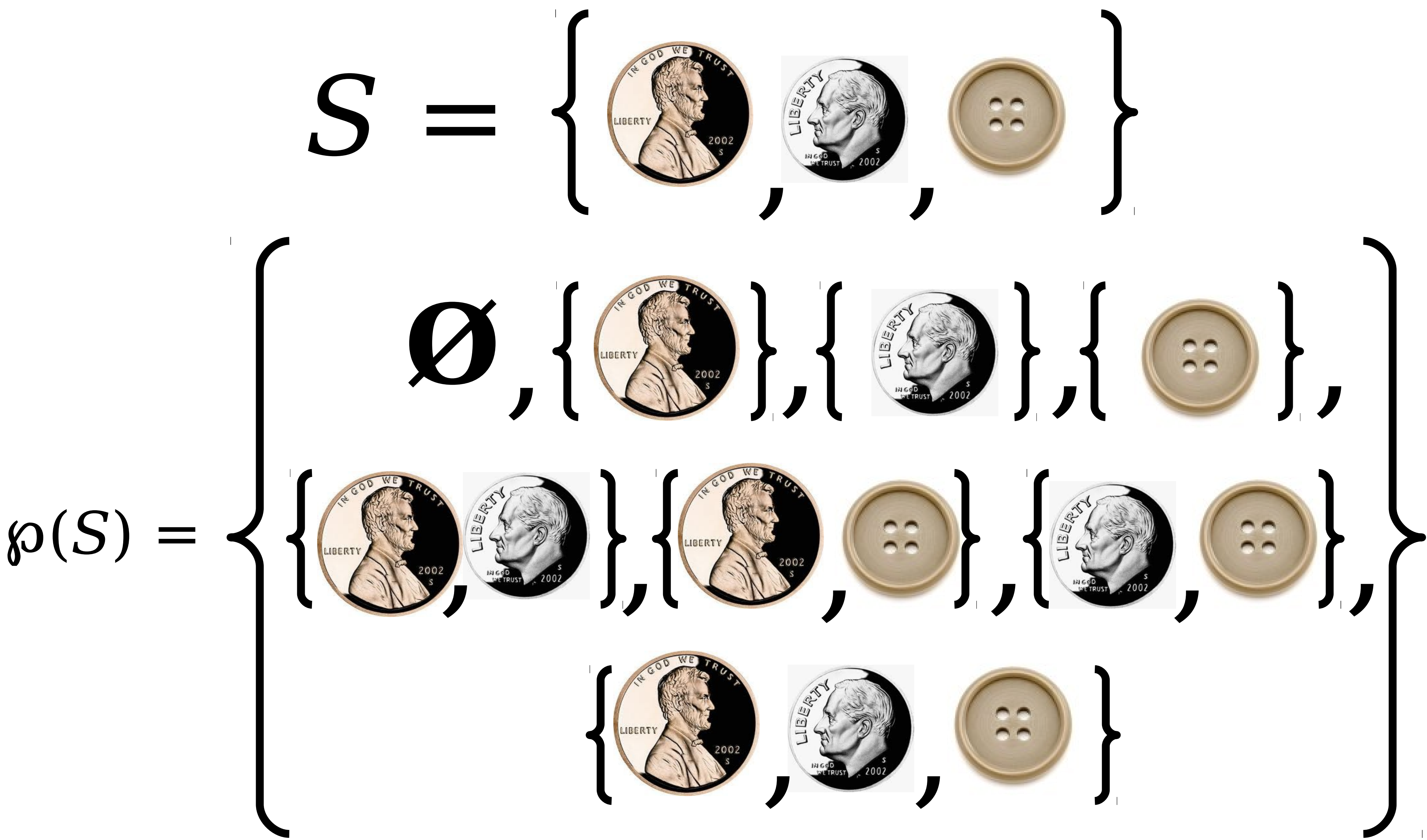
$S = \{$ 

, 

$\}$

$\wp(S) = \{ \varnothing, \{$  $\}, \{$  $\}, \{$  ,  $\} \}$

$$|S| < |\wp(S)|$$



$$|S| < |\wp(S)|$$

$$S = \{a, b, c, d\}$$

$$\wp(S) = \{$$
$$\varnothing,$$
$$\{a\}, \{b\}, \{c\}, \{d\},$$
$$\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}$$
$$\{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\},$$
$$\{a, b, c, d\}$$
$$\}$$

$$\mathbf{|S| < |\wp(S)|}$$

If $|S|$ is infinite, what is the relation between $|S|$ and $|\wp(S)|$?

Does $|S| = |\wp(S)|$?

If $|S| = |\wp(S)|$, we can pair up the elements of S and the elements of $\wp(S)$ without leaving anything out.

If $|S| = |\wp(S)|$, we can pair up the elements of S and the elements of $\wp(S)$ without leaving anything out.

If $|S| = |\wp(S)|$, we can pair up the elements of S and the subsets of S without leaving anything out.

If $|S| = |\wp(S)|$, we can pair up the elements of S and the subsets of S without leaving anything out.

If $|S| = |\wp(S)|$, we can pair up the elements of S and the subsets of S without leaving anything out.

What would that look like?

$$x_0 \longleftrightarrow \{\, x_0,\ x_2,\ x_4,\ \dots \,\}$$

$$x_1 \longleftrightarrow \{\, x_3,\ x_5,\ \dots \,\}$$

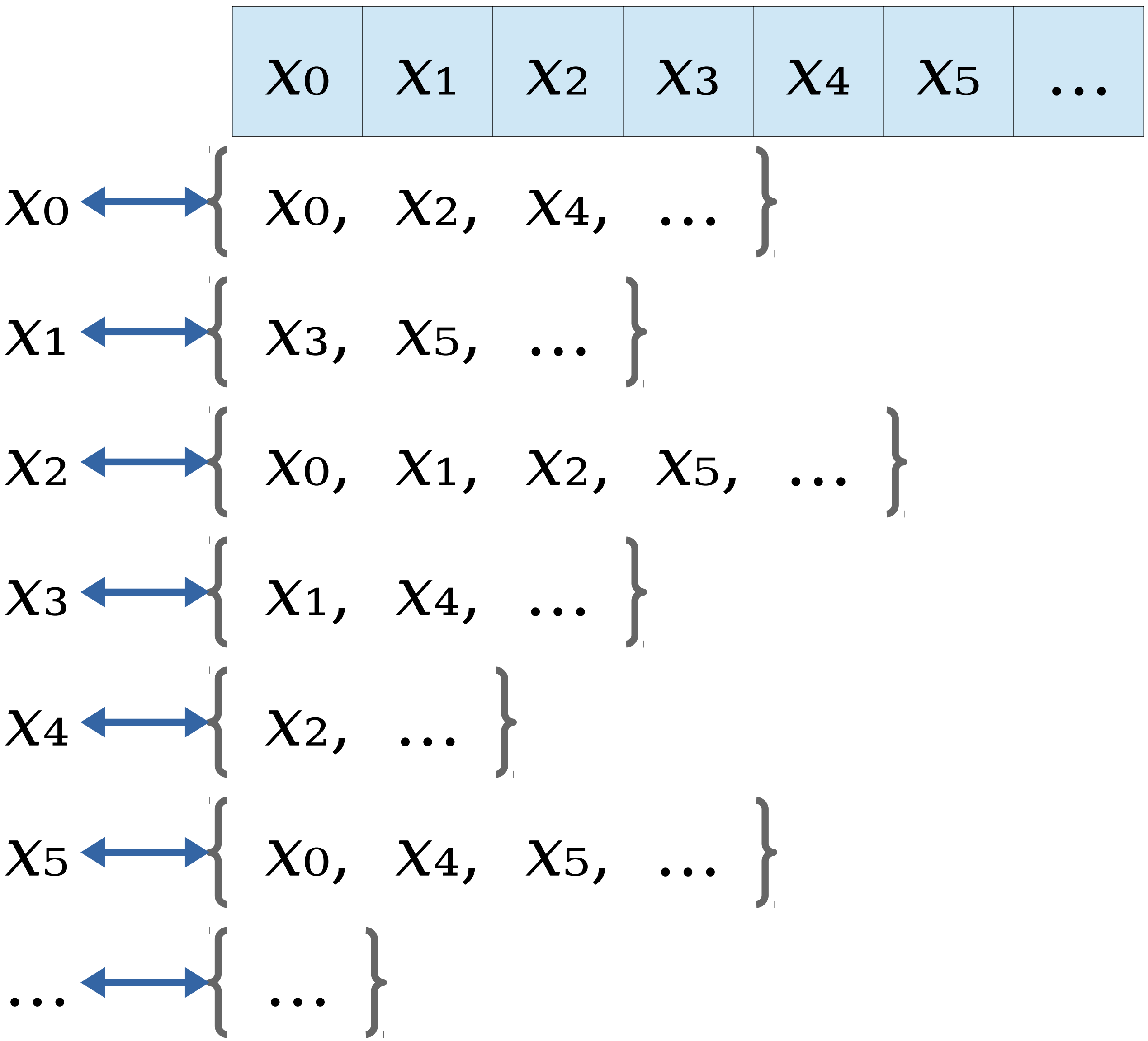
$$x_2 \longleftrightarrow \{\, x_0,\ x_1,\ x_2,\ x_5,\ \dots \,\}$$

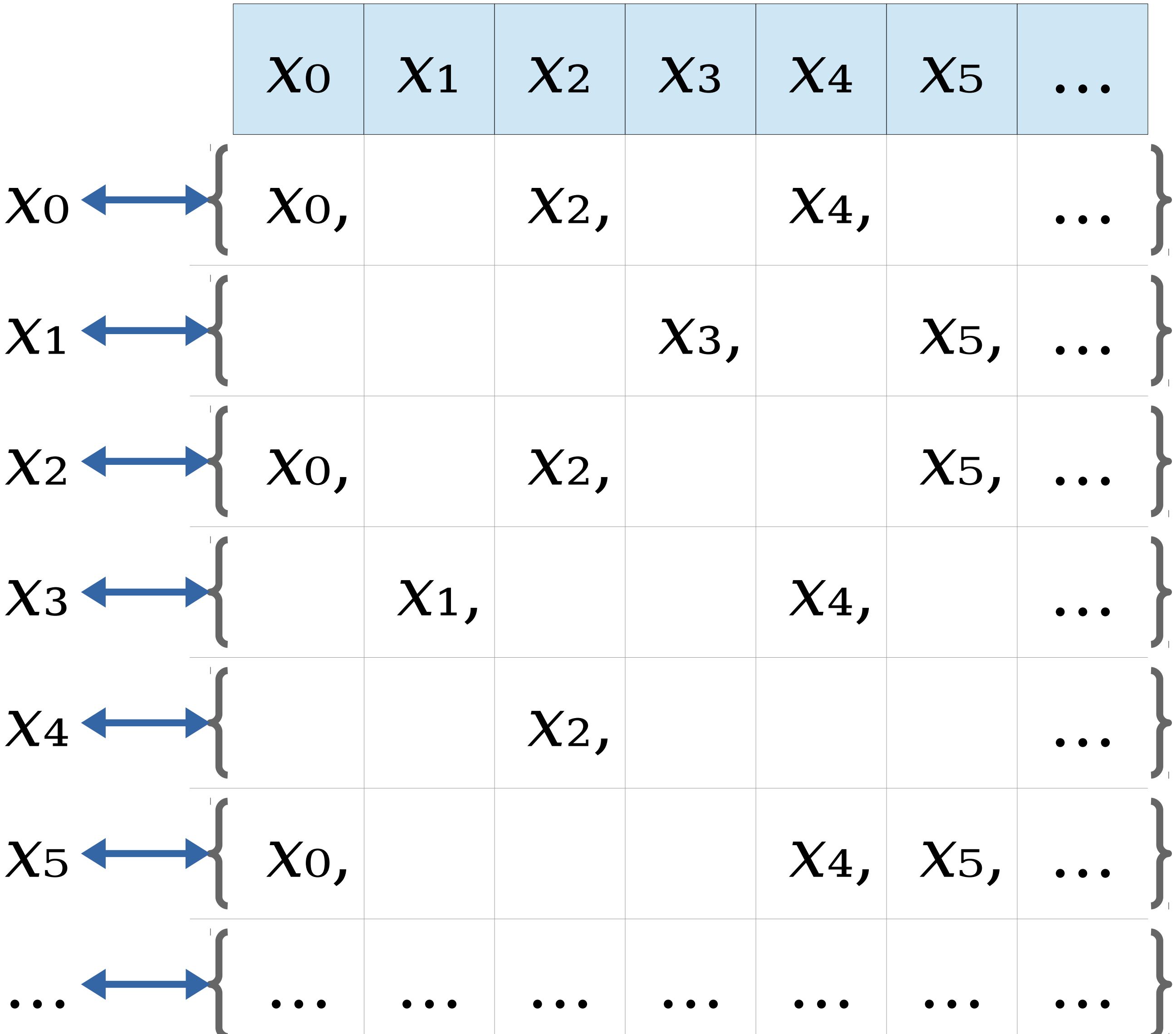
$$x_3 \longleftrightarrow \{\, x_1,\ x_4,\ \dots \,\}$$

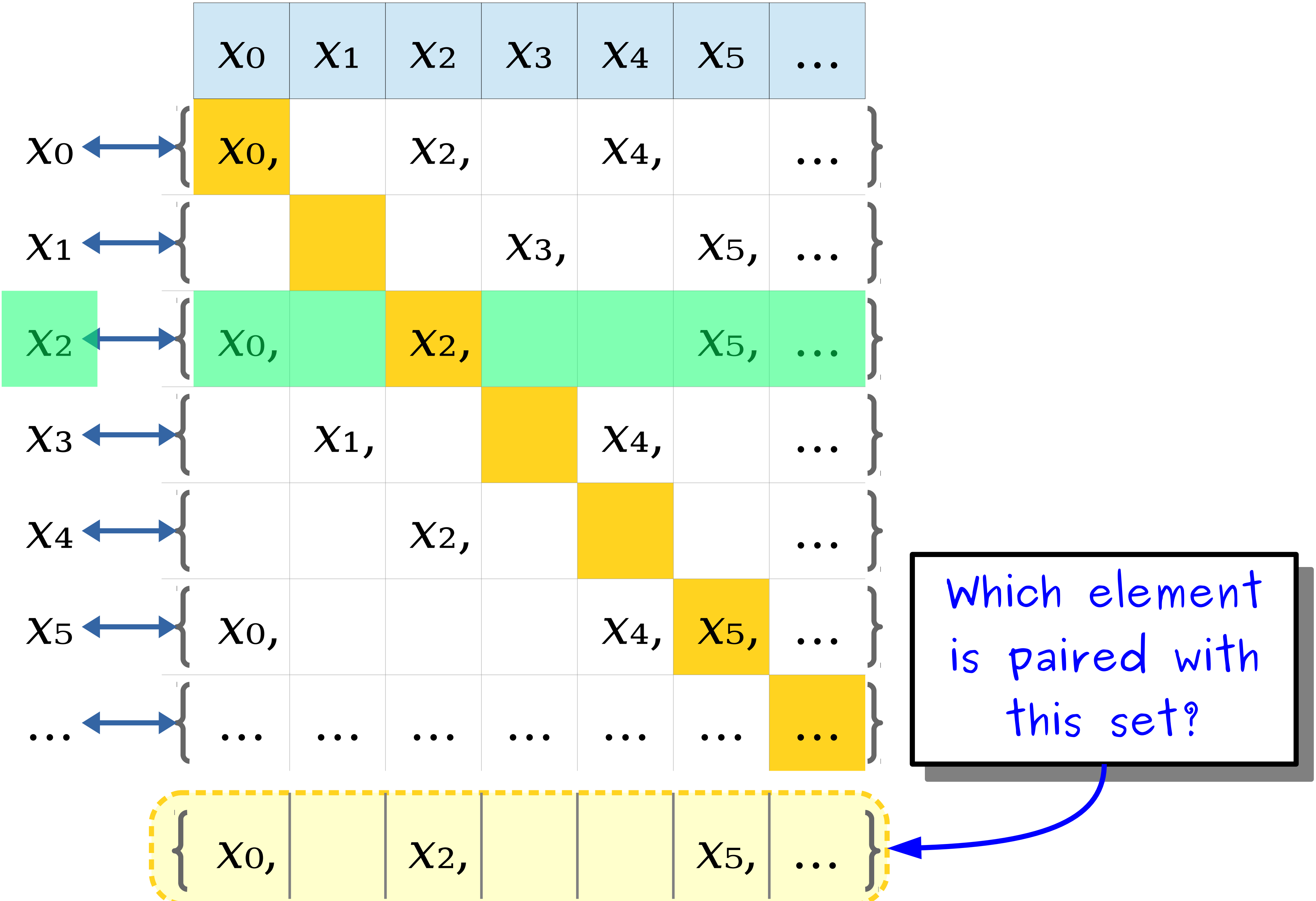
$$x_4 \longleftrightarrow \{\, x_2,\ \dots \,\}$$

$$x_5 \longleftrightarrow \{\, x_0,\ x_4,\ x_5,\ \dots \,\}$$

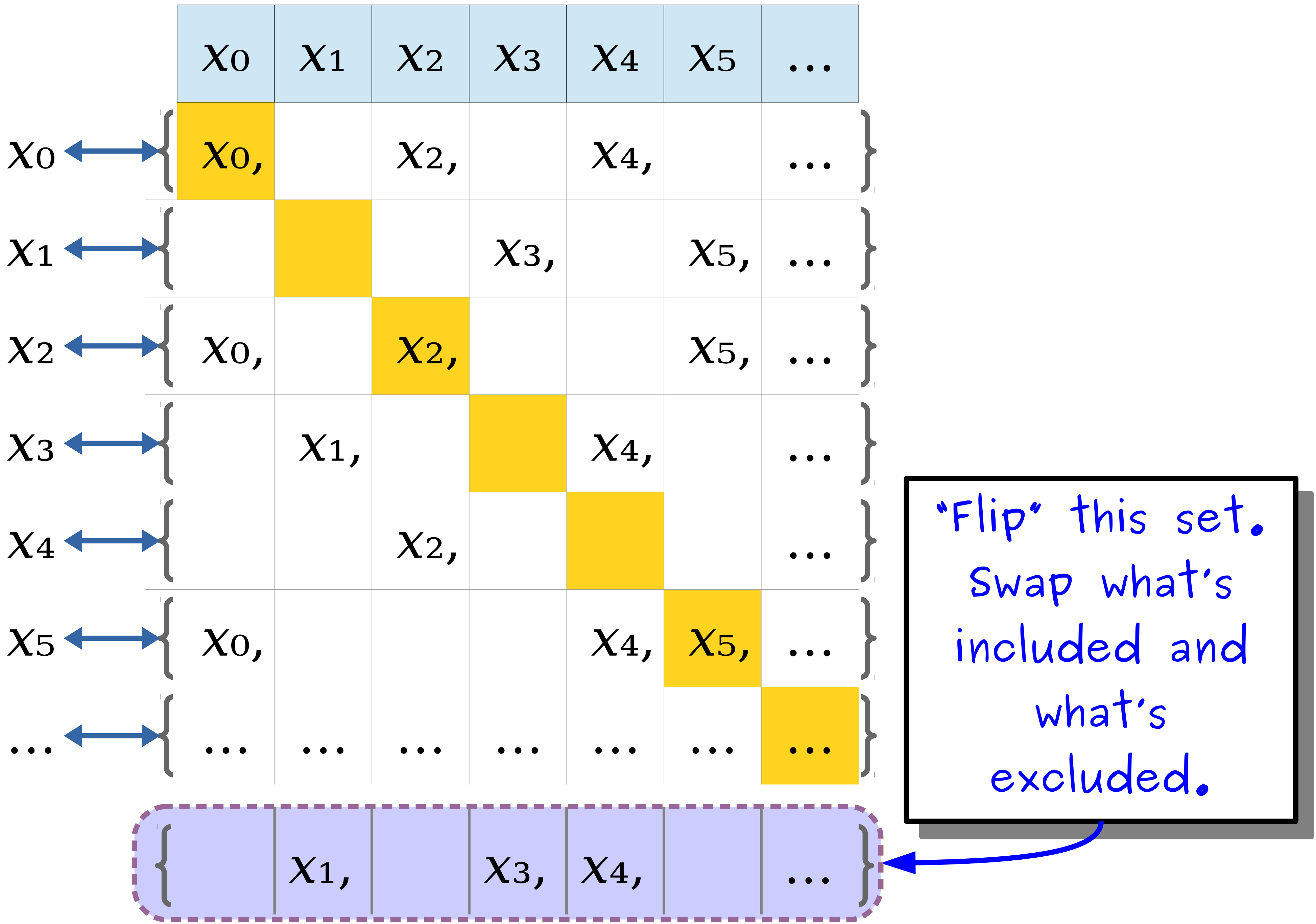
$$\dots \longleftrightarrow \{\, \dots \,\}$$


x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x1, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... }

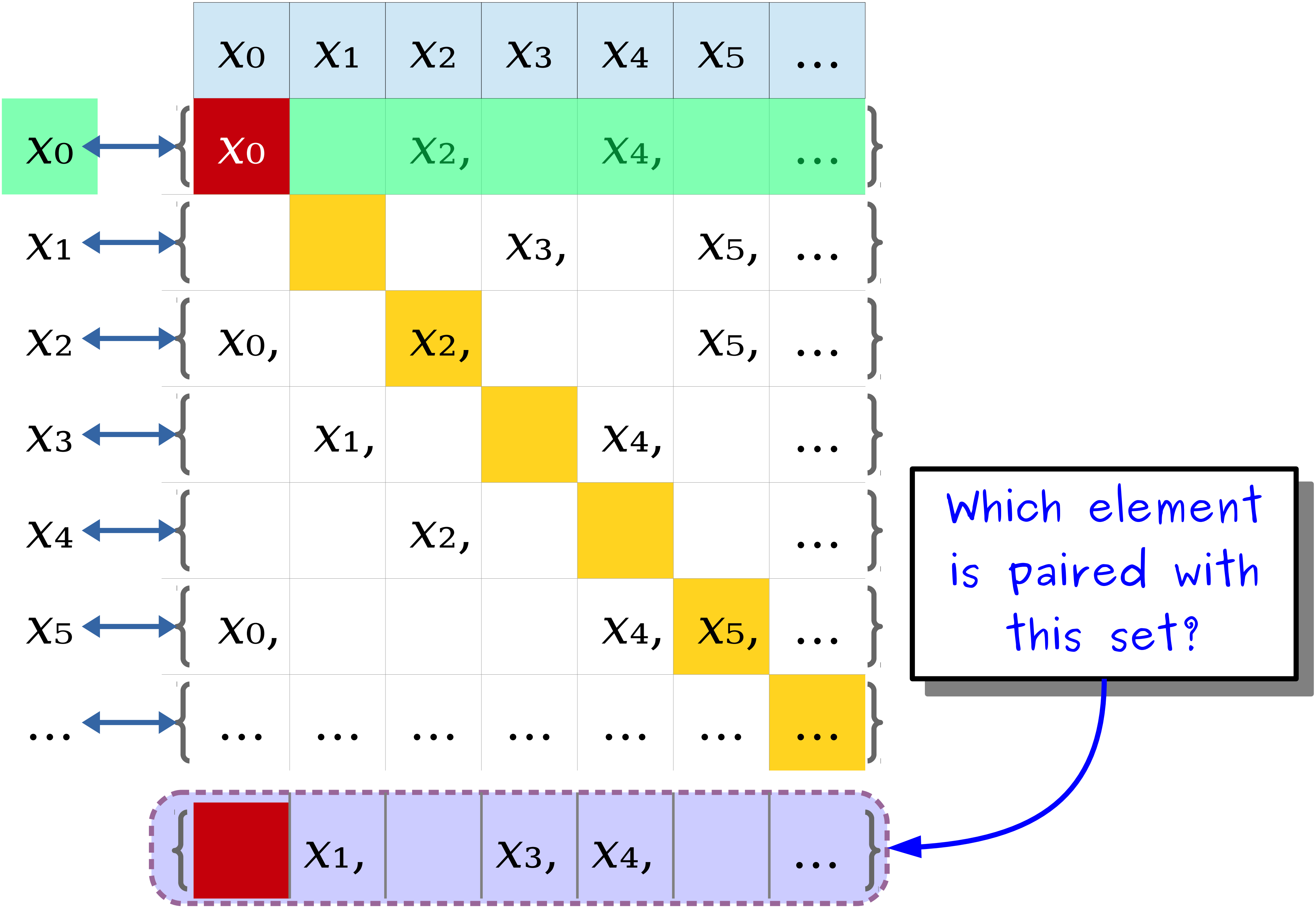

x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... ... }



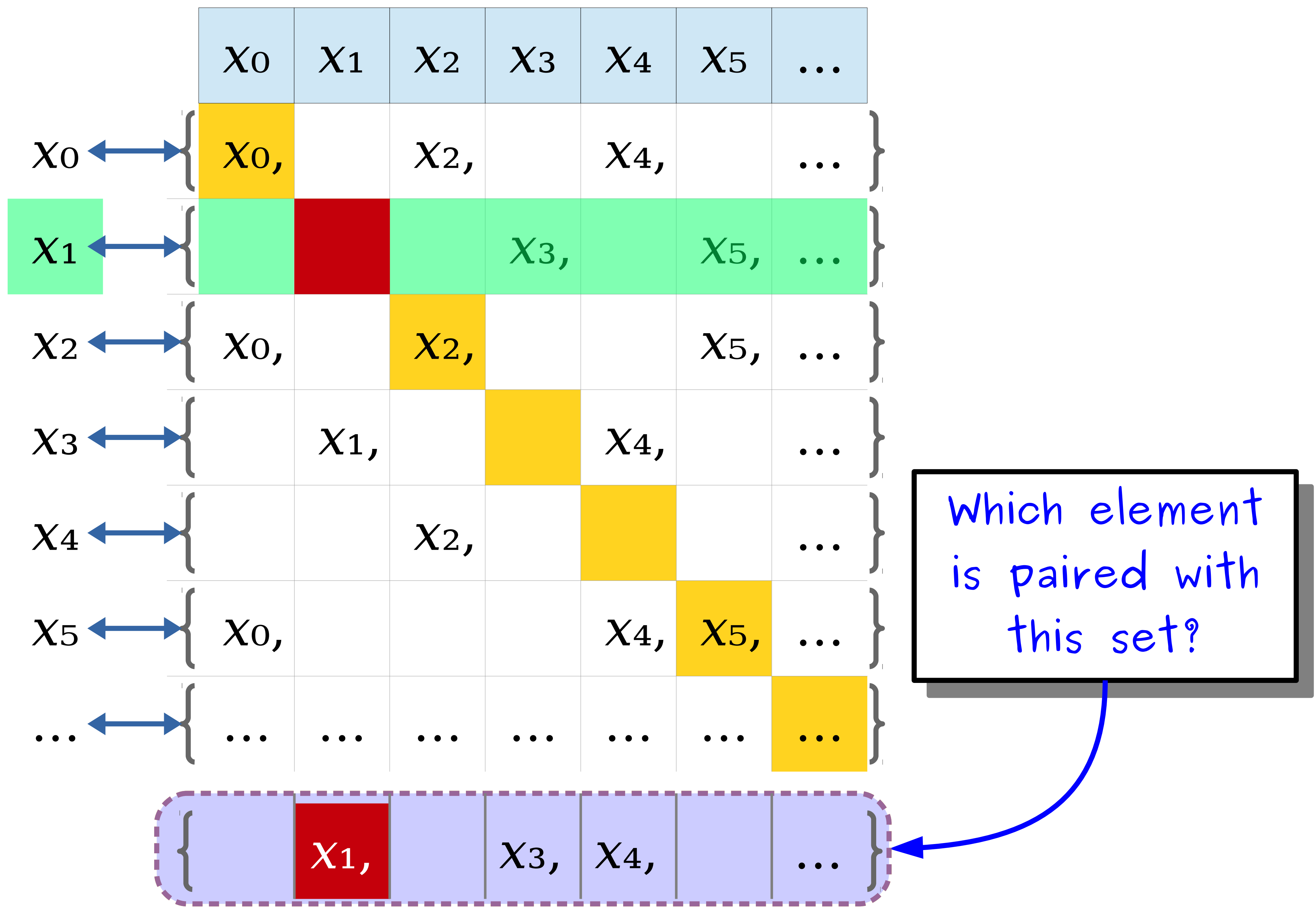
x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... ... }
{ x0, x2, x5, ... }
Which element is paired with this set?



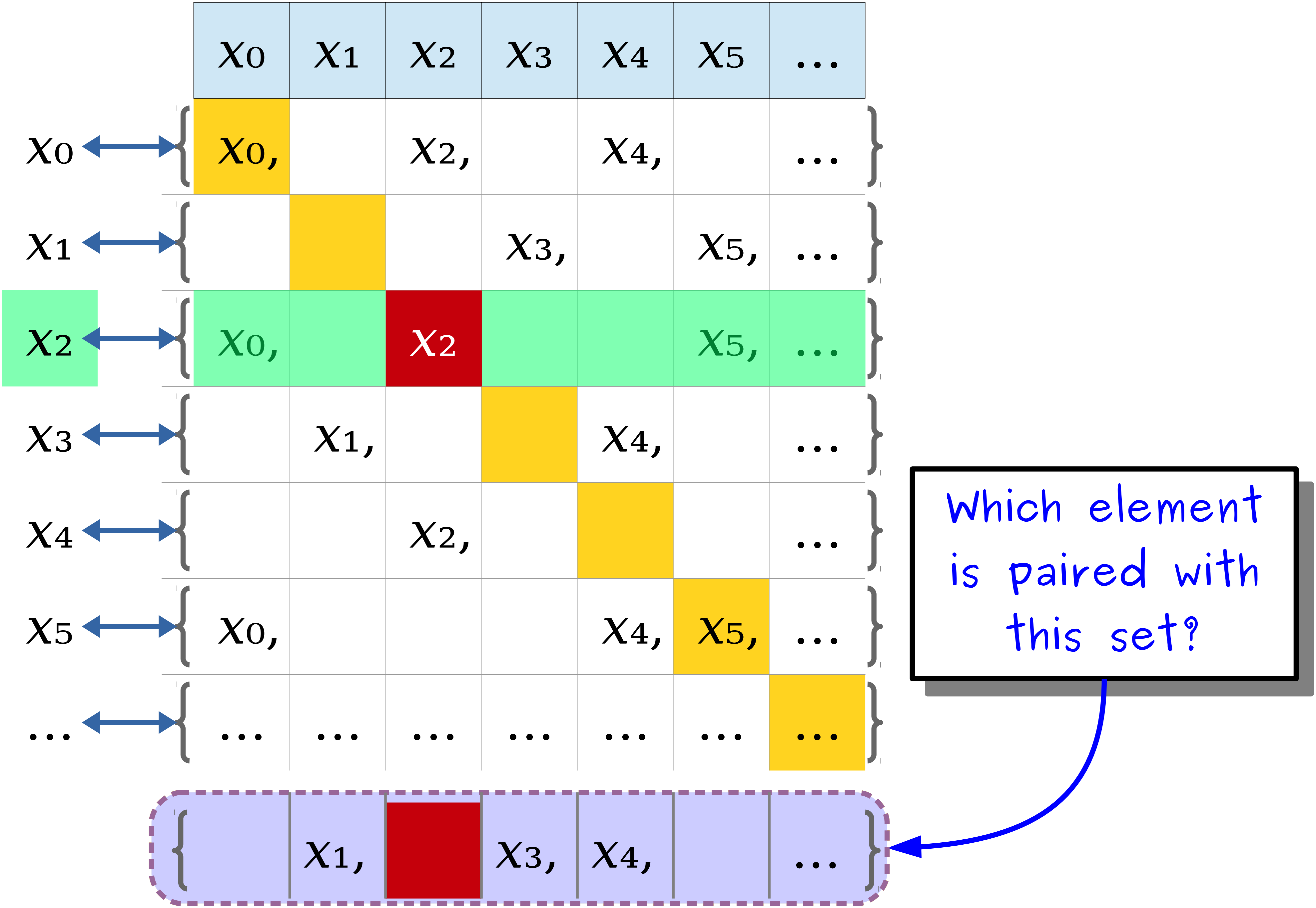
x_0 x_1 x_2 x_3 x_4 x_5 ...
x_0 ↔ { x_0, x_2, x_4, ... }
x_1 ↔ { x_3, x_5, ... }
x_2 ↔ { x_0, x_2, x_5, ... }
x_3 ↔ { x_1, x_4, ... }
x_4 ↔ { x_2, ... }
x_5 ↔ { x_0, x_4, x_5, ... }
... ↔ { ... ... }
{ x_1, x_3, x_4, ... }
"Flip" this set. Swap what's included and what's excluded.



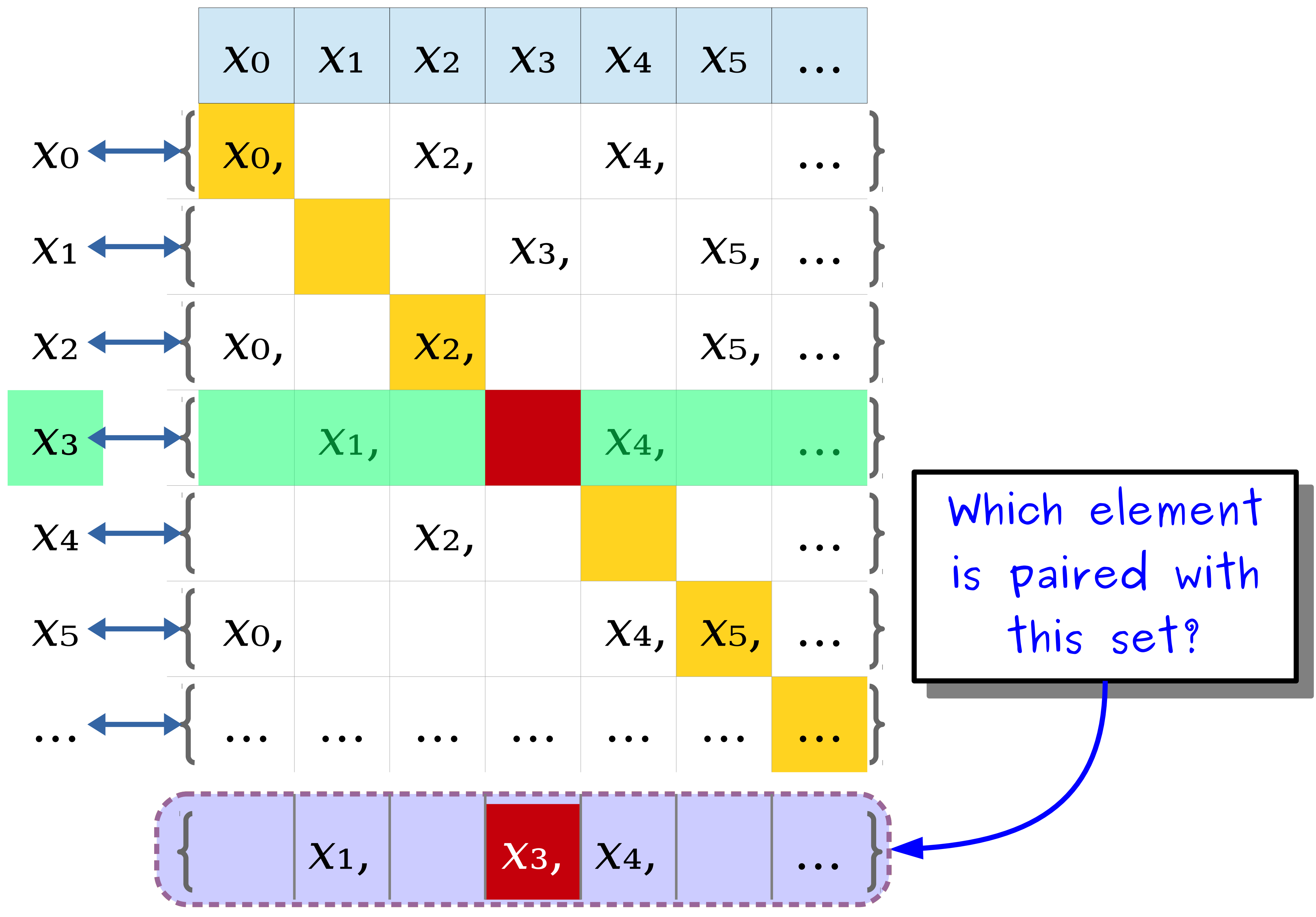
x0 x1 x2 x3 x4 x5 ...
x0 { x0 x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... ... }
{ x1, x3, x4, ... }
Which element is paired with this set?



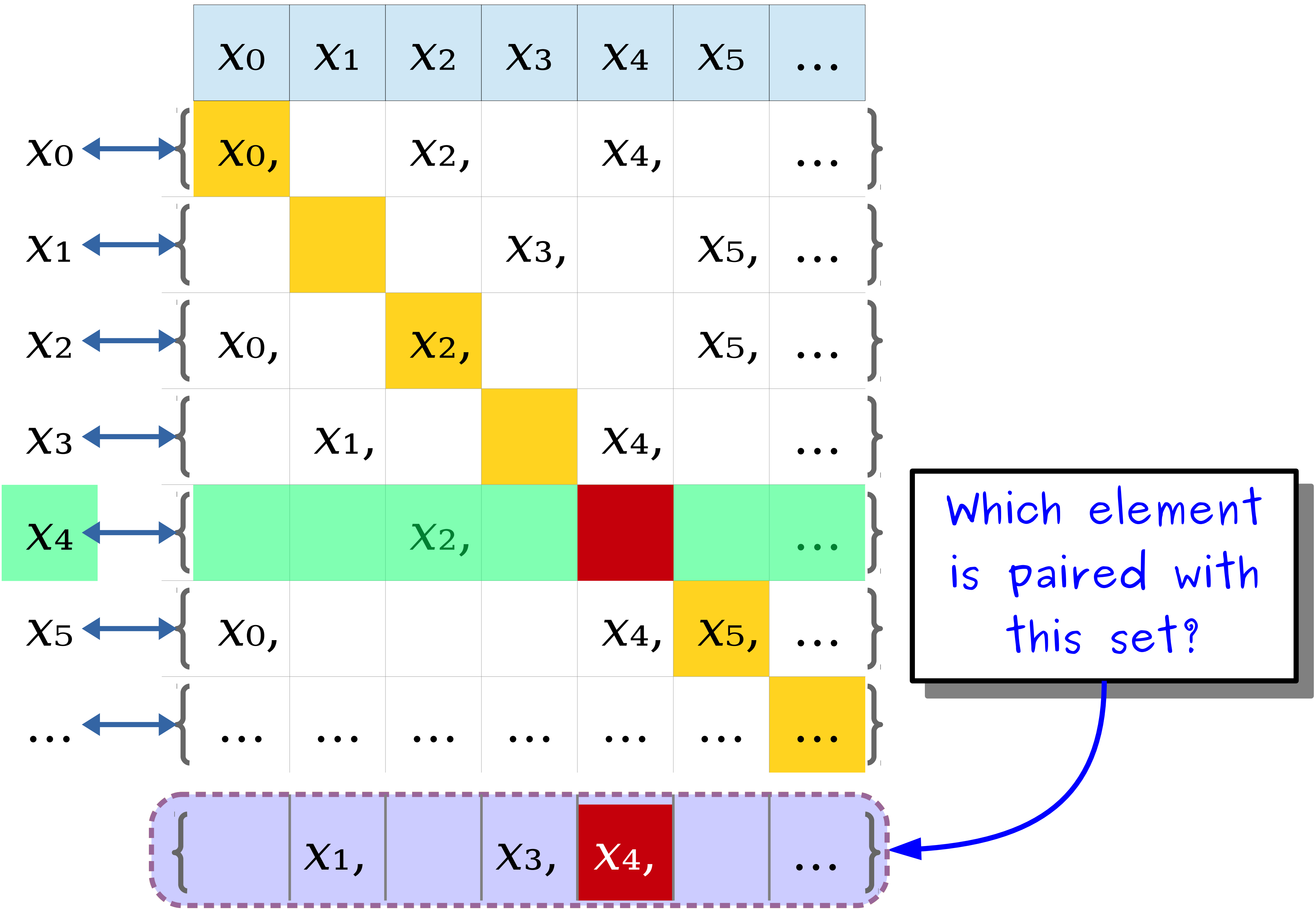
x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... ... }
{ x1, x3, x4, ... }
Which element is paired with this set?



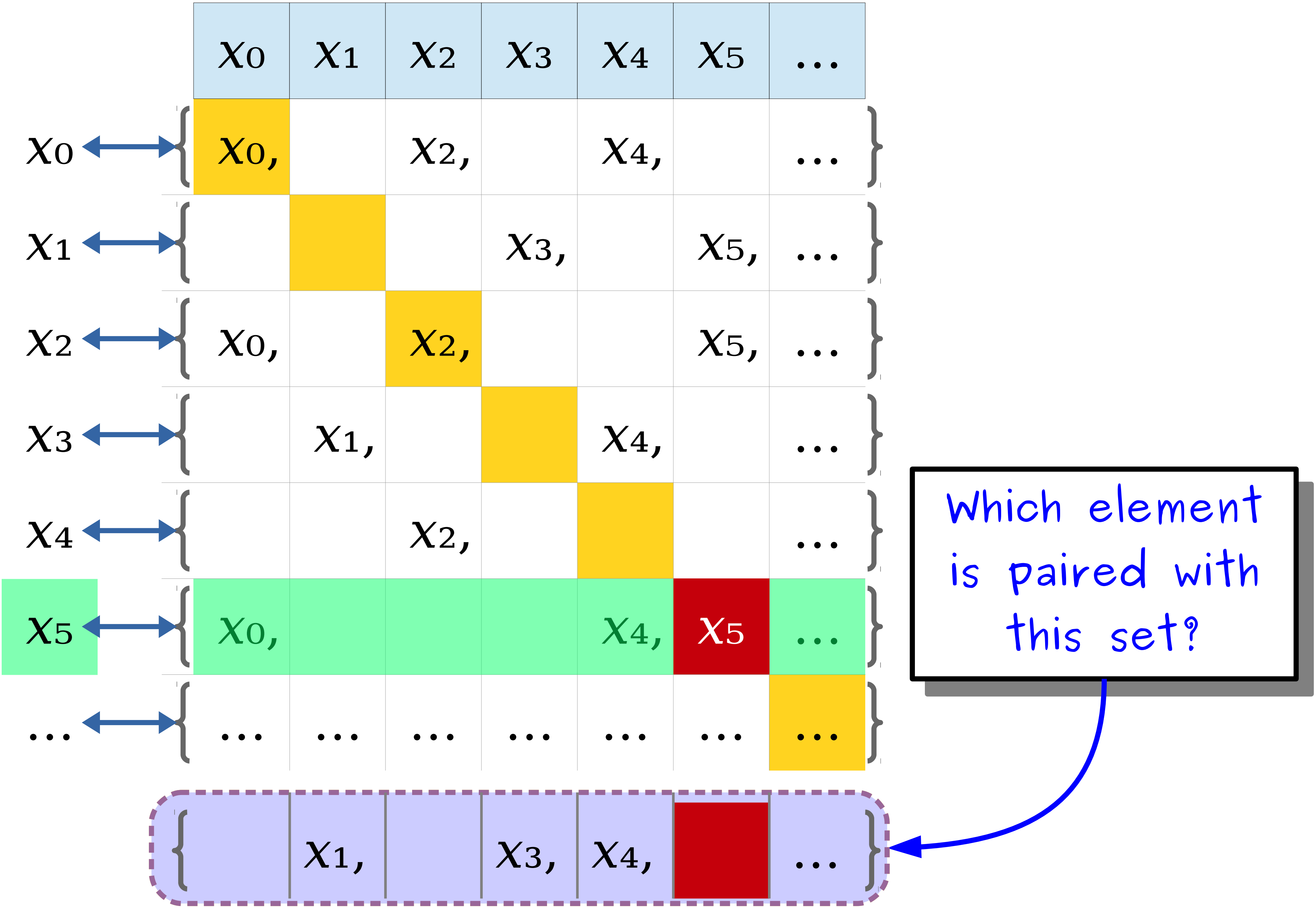
x_0 x_1 x_2 x_3 x_4 x_5 …
x_0 { x_0, x_2, x_4, … }
x_1 { x_3, x_5, … }
x_2 { x_0, x_2 x_5, … }
x_3 { x_1, x_4, … }
x_4 { x_2, … }
x_5 { x_0, x_4, x_5, … }
… { … … … … … … … }
{ x_1, x_3, x_4, … }
Which element is paired with this set?



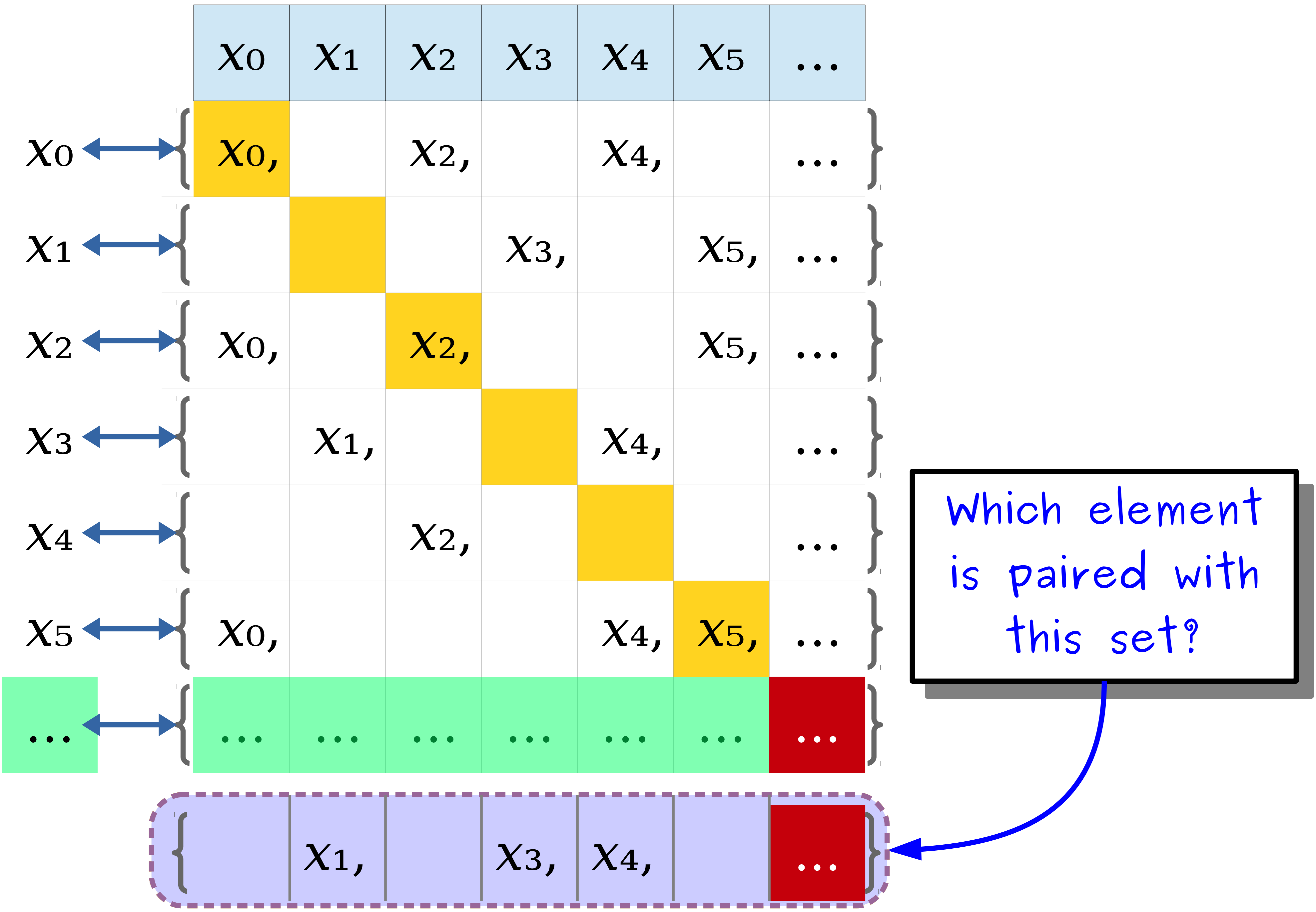
x_0 x_1 x_2 x_3 x_4 x_5 ...
x_0 ↔ { x_0, x_2, x_4, ... }
x_1 ↔ { x_3, x_5, ... }
x_2 ↔ { x_0, x_2, x_5, ... }
x_3 ↔ { x_1, x_4, ... }
x_4 ↔ { x_2, ... }
x_5 ↔ { x_0, x_4, x_5, ... }
... ↔ { ... ... }
{ x_1, x_3, x_4, ... }
Which element is paired with this set?



x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... ... }
{ x1, x3, x4, ... }
Which element is paired with this set?



x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5 ... }
... { ... ... }
{ x1, x3, x4, ... }
Which element is paired with this set?


x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... ... }
{ x1, x3, x4, ... }
Which element is paired with this set?

The Diagonalization Proof

- No matter how we pair up elements of *S* and subsets of *S,* the complemented diagonal won't appear in the table.
  - In row *n,* the *n*th element must be wrong.
- No matter how we pair up elements of *S* and subsets of *S,* there is *always* at least one subset left over.
- This result is ***Cantor's theorem***: Every set is strictly smaller than its power set:

If *S* is a set, then $|S| < |\wp(S)|$.

Infinite Cardinalities

- By Cantor's Theorem:

$$|\mathbb{N}| < |\wp(\mathbb{N})|$$
$$|\wp(\mathbb{N})| < |\wp(\wp(\mathbb{N}))|$$
$$|\wp(\wp(\mathbb{N}))| < |\wp(\wp(\wp(\mathbb{N})))|$$
$$|\wp(\wp(\wp(\mathbb{N})))| < |\wp(\wp(\wp(\wp(\mathbb{N}))))|$$
$$\ldots$$

- ***Not all infinite sets have the same size!***
- ***There is no biggest infinity!***
- ***There are infinitely many infinities!***

What does this have to do with computation?

“The set of all computer programs”

“The set of all problems to solve”

Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings.
  - There are ***at least*** as many problems as there are sets of strings.
- This leads to some *incredible* results – we'll see why in a minute!

Where We're Going

A ***string*** is a sequence of characters.

We're going to prove the following results:

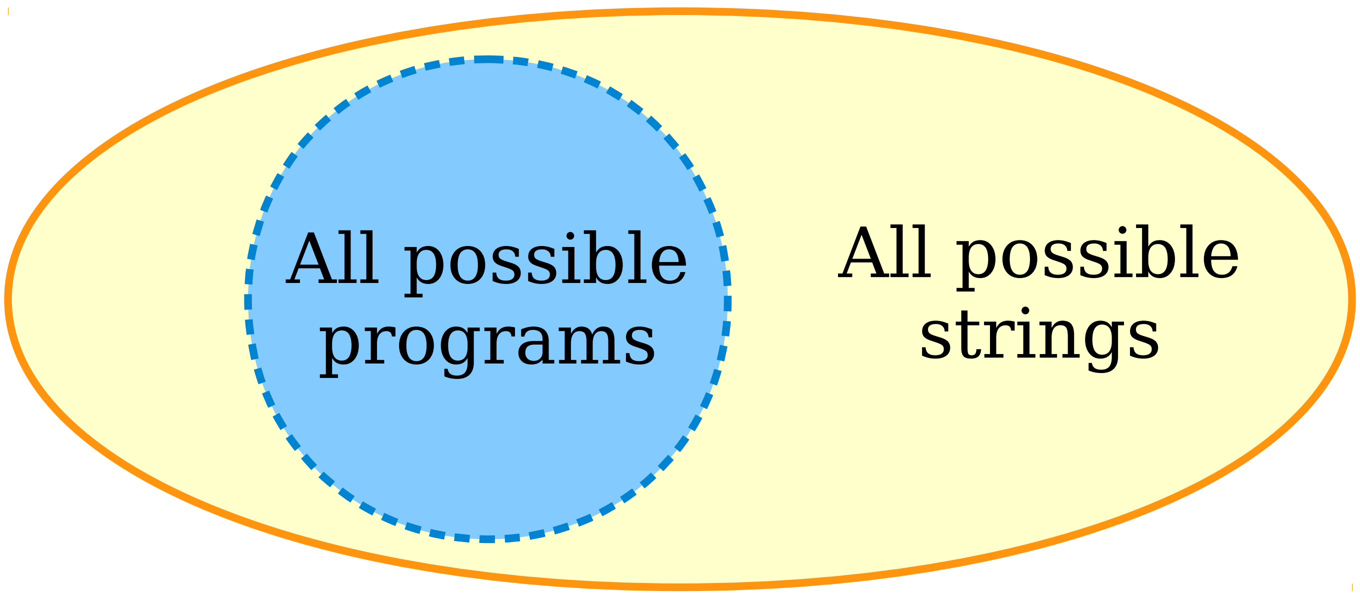
- There are ***at most*** as many programs as there are strings.

  There are ***at least*** as many problems as there are sets of strings.

This leads to some *incredible* results – we'll see why in a minute!

Strings and Programs

- The source code of a computer program is just a (long, structured, well-commented) string of text.
- All programs are strings, but not all strings are necessarily programs.



|Programs| ≤ |Strings|

Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings.
  - There are ***at least*** as many problems as there are sets of strings.
- This leads to some *incredible* results – we'll see why in a minute!

Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings. ✓
  - There are ***at least*** as many problems as there are sets of strings.
- This leads to some *incredible* results – we'll see why in a minute!

Where We're Going

A ***string*** is a sequence of characters.

We're going to prove the following results:

- There are ***at most*** as many programs as there are strings. ✓
- There are ***at least*** as many problems as there are sets of strings.

This leads to some *incredible* results – we'll see why in a minute!

Strings and Problems

- There is a connection between the number of sets of strings and the number of problems to solve.
- Let S be any set of strings. This set S gives rise to a problem to solve:

Given a string w, determine whether $w \in S$.

Strings and Problems

Given a string w, determine whether $w \in S$.

- Suppose that S is the set

$$S = \{ \text{"}a\text{"}, \text{"}b\text{"}, \text{"}c\text{"}, \ldots, \text{"}z\text{"} \}$$

- From this set S, we get this problem:

Given a string w, determine whether w is a single lower-case English letter.

Strings and Problems

Given a string *w*, determine whether $w \in S$.

- Suppose that S is the set

$$S = \{ \text{"0"}, \text{"1"}, \text{"2"}, \ldots, \text{"9"}, \text{"10"}, \text{"11"}, \ldots \}$$

- From this set S, we get this problem:

Given a string *w*, determine whether *w* represents a natural number.

Strings and Problems

Given a string *w*, determine whether $w \in S$.

- Suppose that S is the set

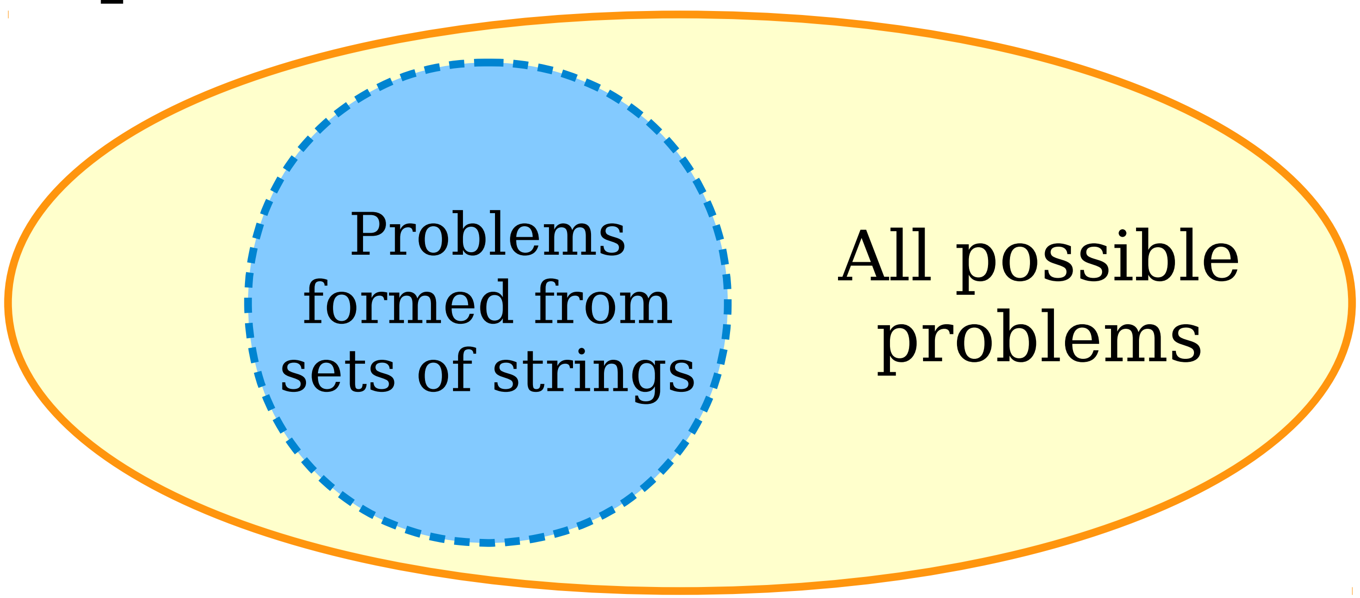
$$S = \{ p \mid p \text{ is a legal C++ program } \}$$

- From this set S, we get this problem:

Given a string *w*, determine whether *w* is a legal C++ program.

Strings and Problems

- Every set of strings gives rise to a unique problem to solve.
- Other problems exist as well.



|Sets of Strings| ≤ |Problems|

Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings. ✓
  - There are ***at least*** as many problems as there are sets of strings.
- This leads to some *incredible* results – we'll see why in a minute!

Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings. ✓
  - There are ***at least*** as many problems as there are sets of strings. ✓
- This leads to some *incredible* results – we'll see why in a minute!

Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings. ✓
  - There are ***at least*** as many problems as there are sets of strings. ✓
- This leads to some *incredible* results – we'll see why in a minute!

Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings. ✓
  - There are ***at least*** as many problems as there are sets of strings. ✓
- This leads to some *incredible* results – we'll see why ~~in a minute!~~ *right now!*

Every computer program is a string.

So, the number of programs is at most the number of strings.

From Cantor's Theorem, we know that there are more sets of strings than strings.

There are at least as many problems as there are sets of strings.

$$|\textbf{Programs}| \leq |\textbf{Strings}| < |\wp(\textbf{Strings})| \leq |\textbf{Problems}|$$

Every computer program is a string.

So, the number of programs is at most the number of strings.

From Cantor's Theorem, we know that there are more sets of strings than strings.

There are at least as many problems as there are sets of strings.

|Programs| < |Problems|

There are more problems to solve than there are programs to solve them.

|Programs| < |Problems|

It Gets Worse

- Using more advanced set theory, we can show that there are *infinitely more* problems than solutions.
- In fact, if you pick a totally random problem, the probability that you can solve it is *zero*.
- ***More troubling fact:*** We've just shown that *some* problems are impossible to solve with computers, but we don't know *which* problems those are!

We need to develop a more nuanced understanding of computation.

Where We're Going

- ***What makes a problem impossible to solve with computers?***
  - Is there a deep reason why certain problems can't be solved with computers, or is it completely arbitrary?
  - How do you know when you're looking at an impossible problem?
  - Are these real-world problems, or are they highly contrived?
- ***How do we know that we're right?***
  - How can we back up our pictures with rigorous proofs?
  - How do we build a mathematical framework for studying computation?

Next Time

- ***Mathematical Proof***
  - What is a mathematical proof?
  - How can we prove things with certainty?