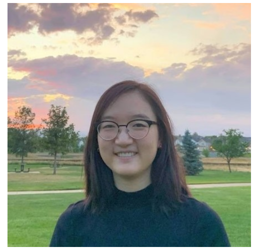
Welcome to CS103!

Are there “laws of physics” in computer science?

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***

Amy Liu
(Instructor)



Jennie Chung
(Head TA)

Ryan Guan
(ACE Instructor)

Benson Kung

Deven Pandya

Edgar Roman

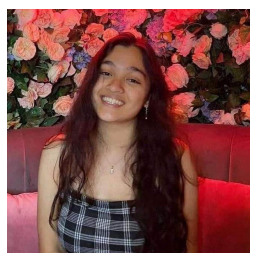
Kanoe Aiu

Lucas Bosman

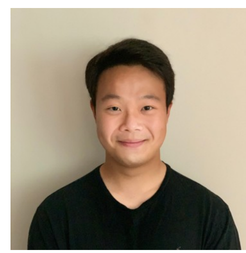
Matthew Sotoudeh

Mona Aanvarihosseinabad

Rachel Han



Reva Agashe

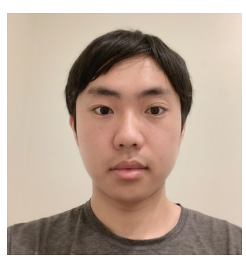


Stanley Cao

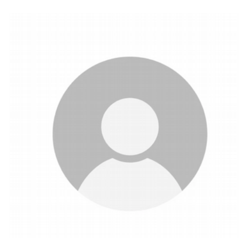
Tracy Chang

Vyoma Raman

Wanyue Zhai



Warren Xia



Yubin Lee

Staff Email List: ***cs103-win2324-staff@lists.stanford.edu***

Course Website

https://cs103.stanford.edu

All course content will be hosted here, except for lecture videos.

Prerequisite / Corequisite

CS106B

Some problem sets will have small coding components. We'll also reference some concepts from CS106B, particularly recursion, throughout the quarter.

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

Another Option

CS154

CS154 is more appropriate if you have a background in the topics from the first half of this quarter (set theory, proofwriting, discrete math, formal logic, graphs, etc.) Come talk to me after class if you're curious about this!

CS103 ACE

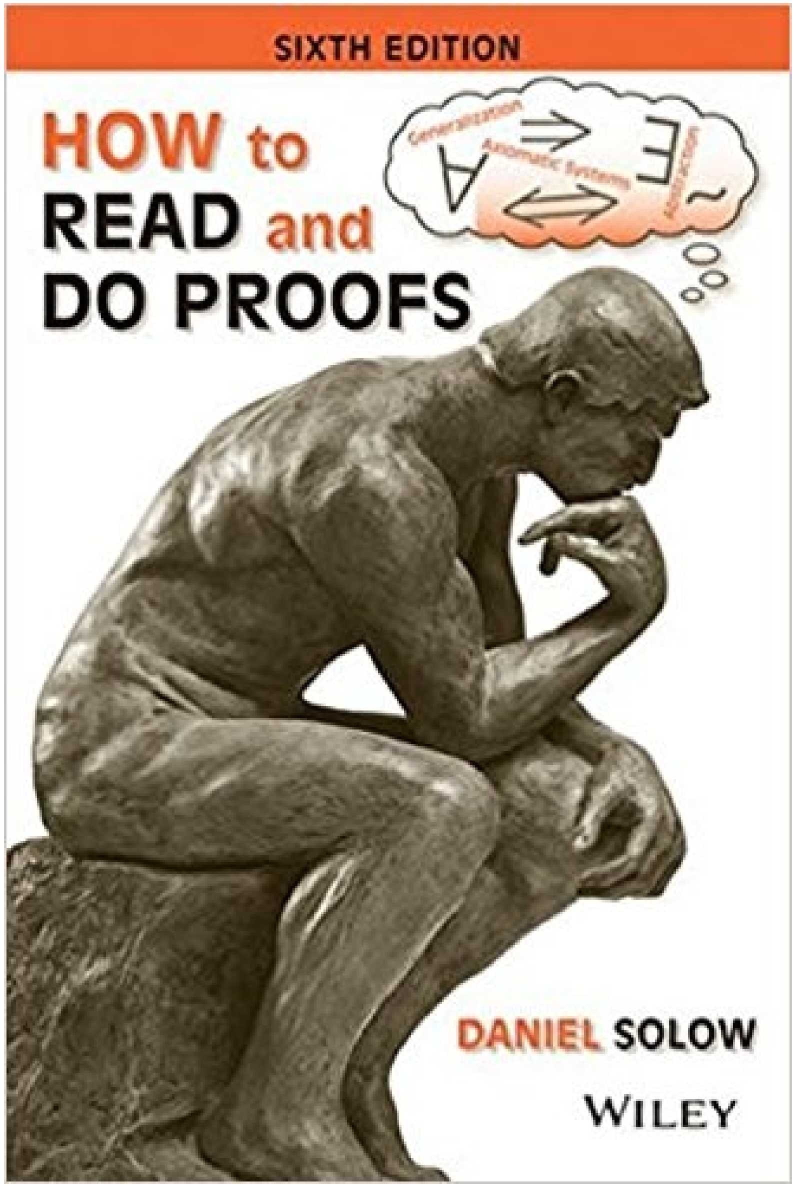
- ***CS103 ACE*** is an optional, one-unit companion course to CS103.
- CS103 ACE meets Tuesdays from 3:00PM – 4:50PM and provides additional practice with the course material in a small group setting.
- The first course meeting is next week.
- Interested? Apply online using ***this link***.
- The CS103 ACE materials are available to everyone. You can pull them up ***here***.

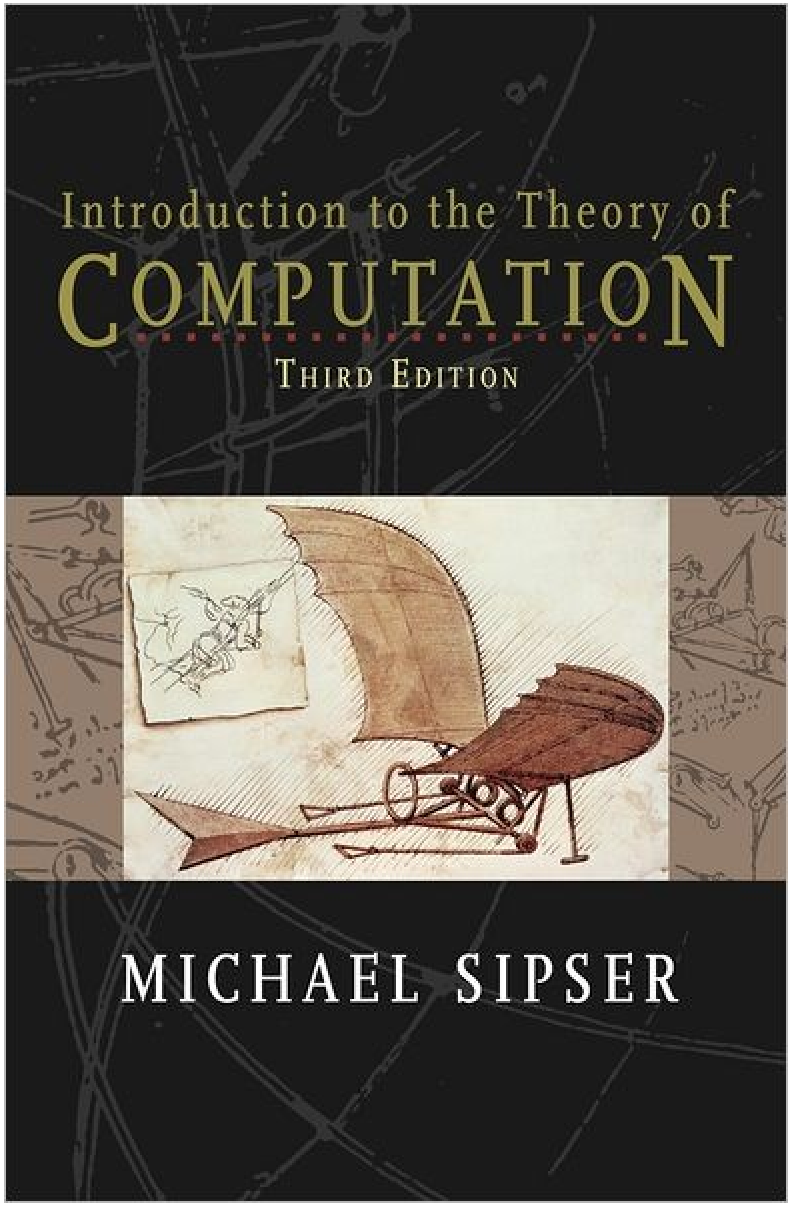
Ryan Guan
(ACE Instructor)

Problem Set 0

- Your first assignment, Problem Set 0, goes out today. It’s due Friday at 4:00PM Pacific.
- This assignment requires you to set up your development environment and to get set up on GradeScope.
- There’s no coding involved, but it’s good to start early in case you encounter any technical issues.

Recommended Reading





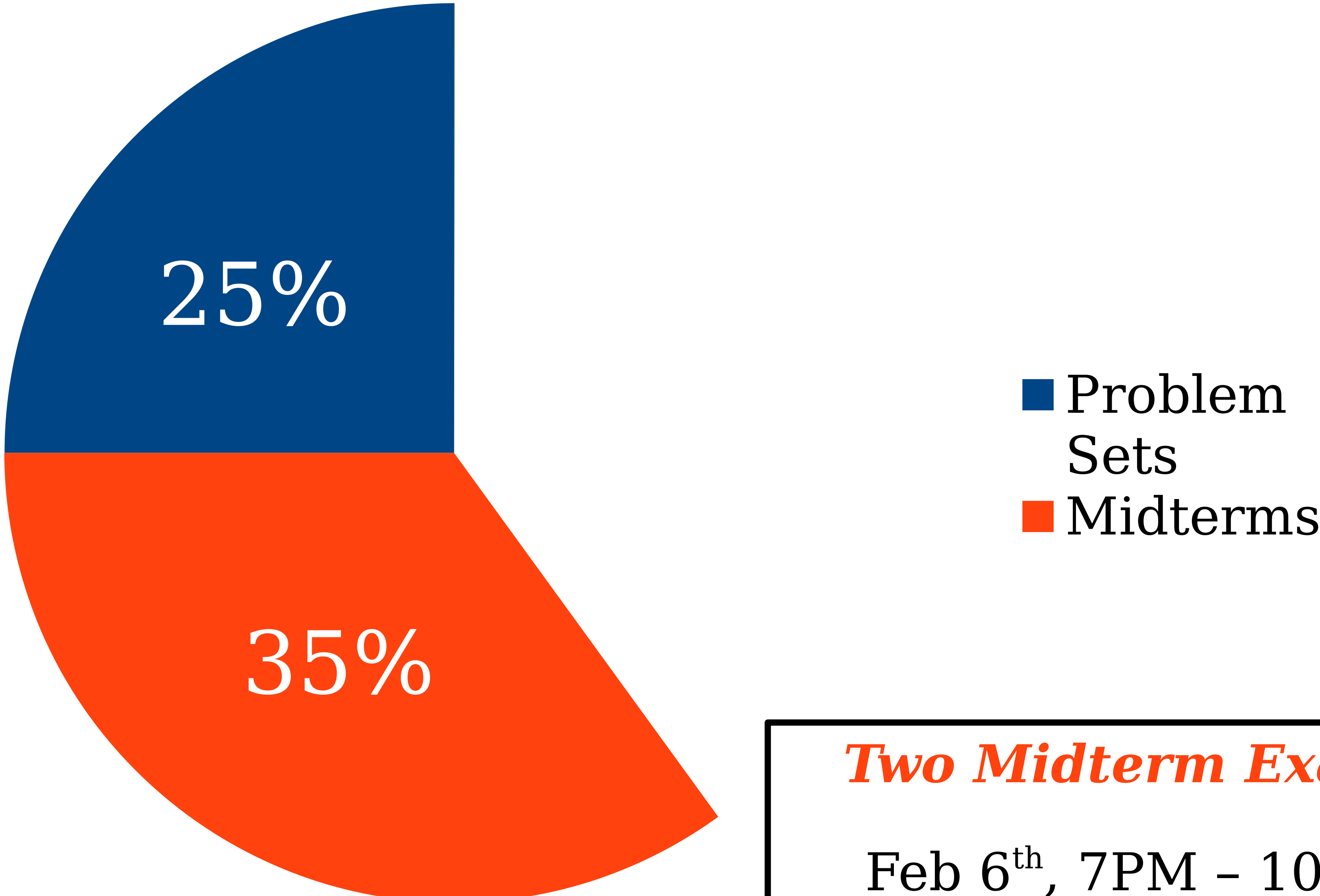
Grading

Grading

Ten Problem Sets

Completed individually or in pairs.

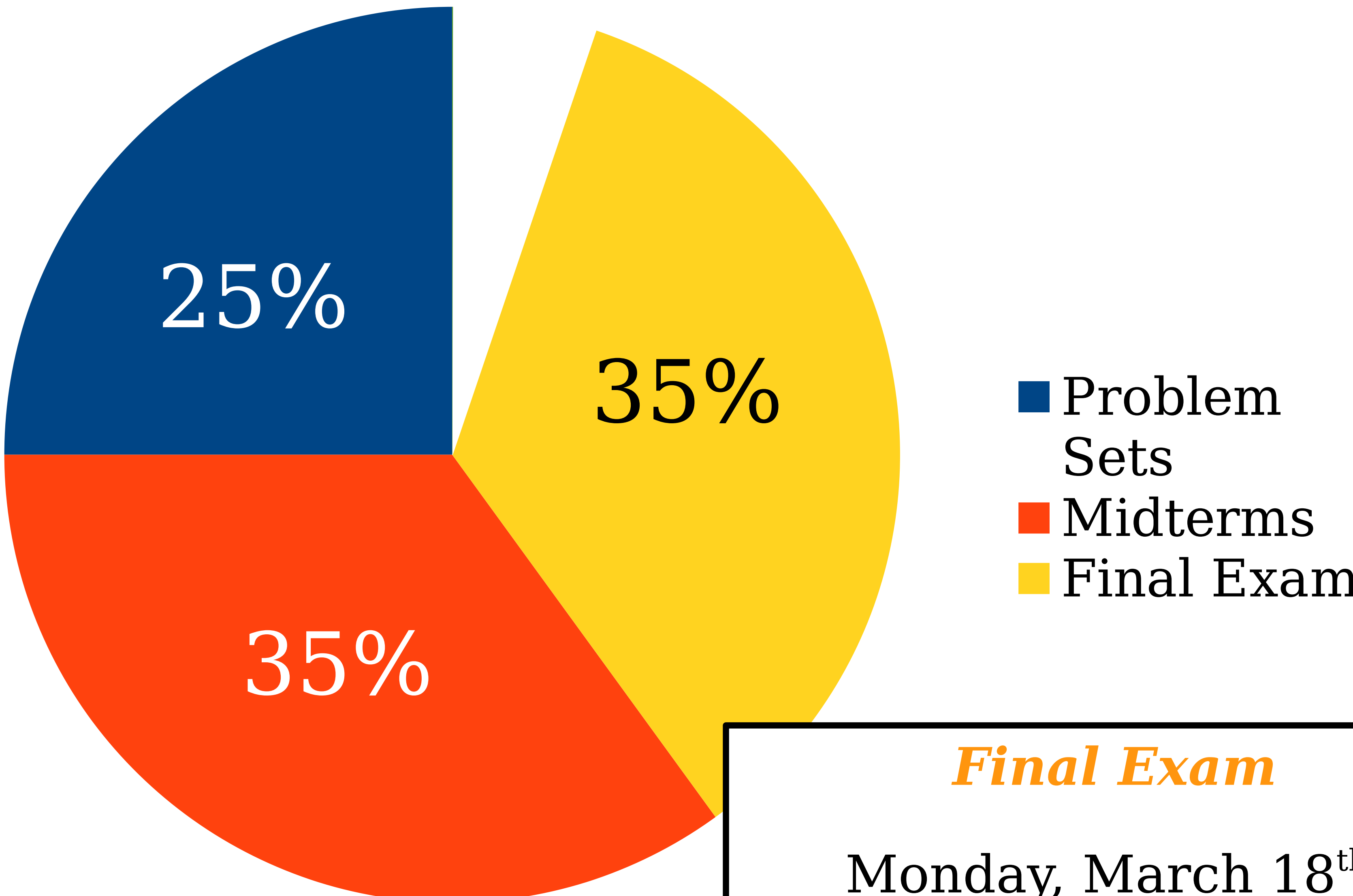
Grading



Two Midterm Exams

Feb 6th, 7PM – 10PM
Feb 27th, 7PM – 10PM

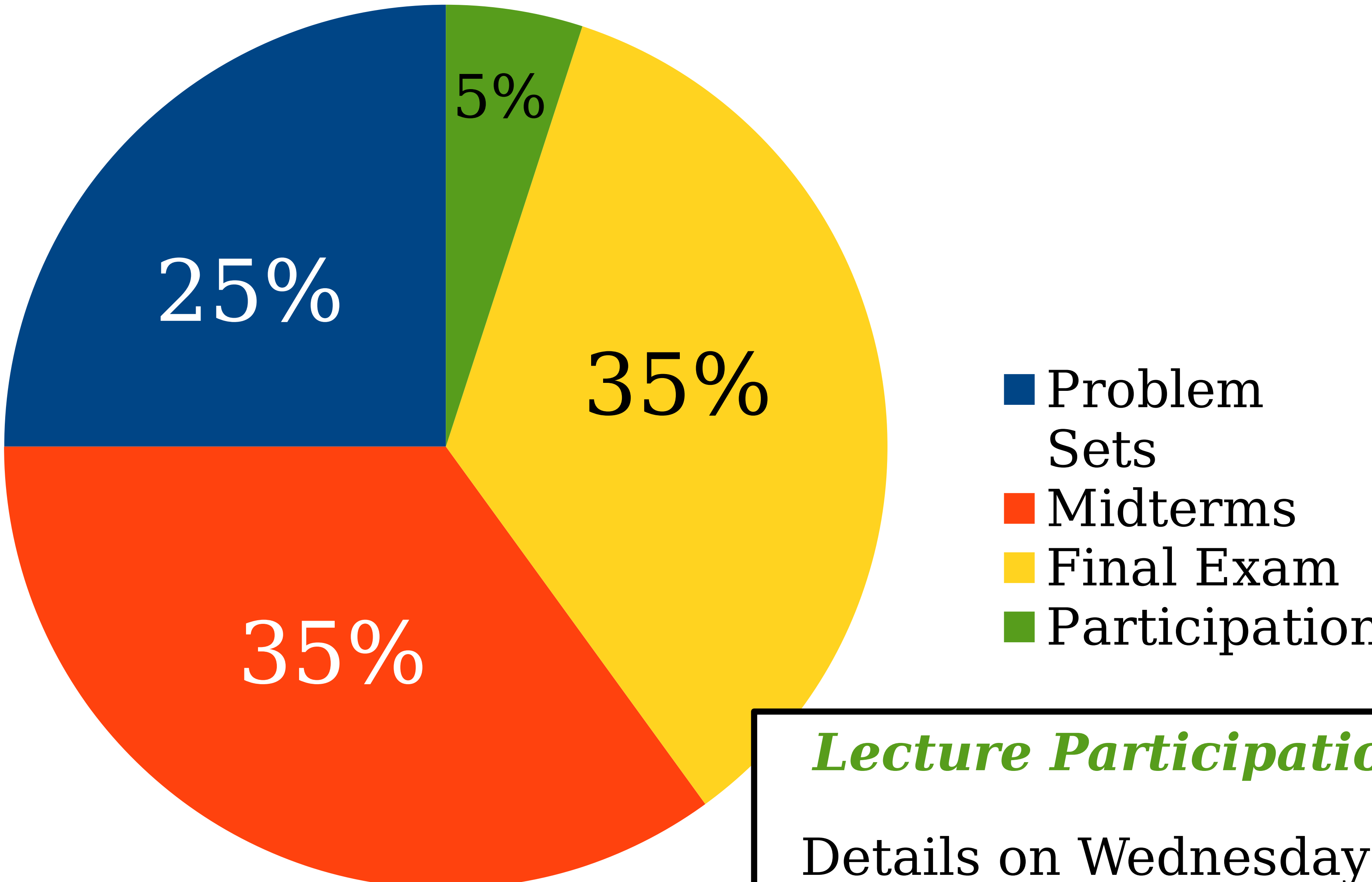
Grading



Final Exam

Monday, March 18th, 7PM – 10PM.

Grading



Lecture Participation

Details on Wednesday of next week!

We've got a big journey ahead of us.

Let's get started!

Introduction to Set Theory

"CS103 students"

"Cool people"

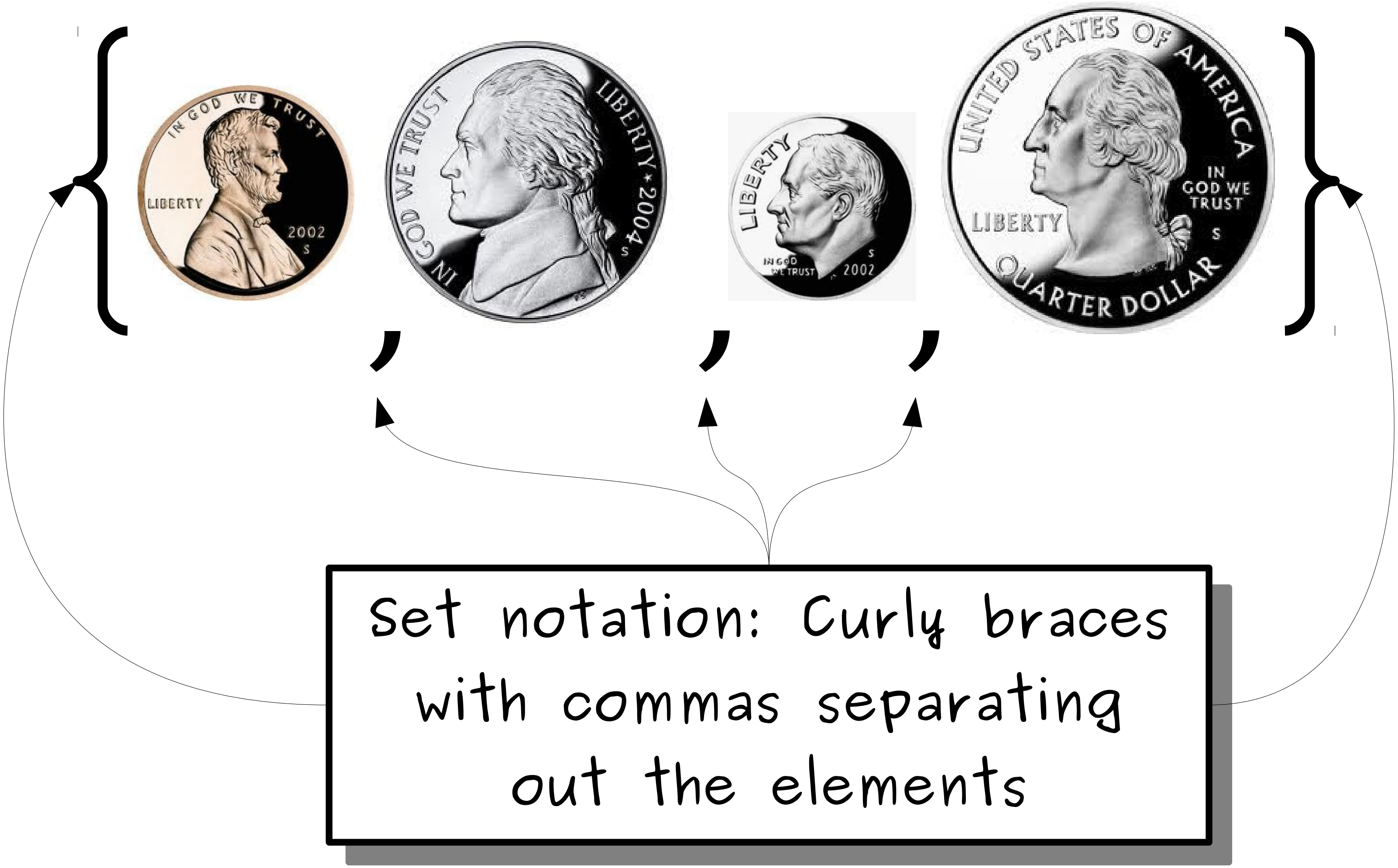
"The chemical elements"

"Cute animals"

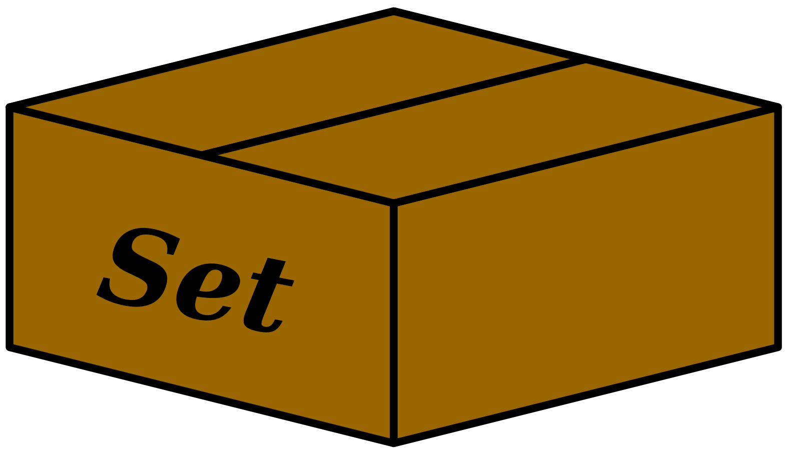
"US coins"

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

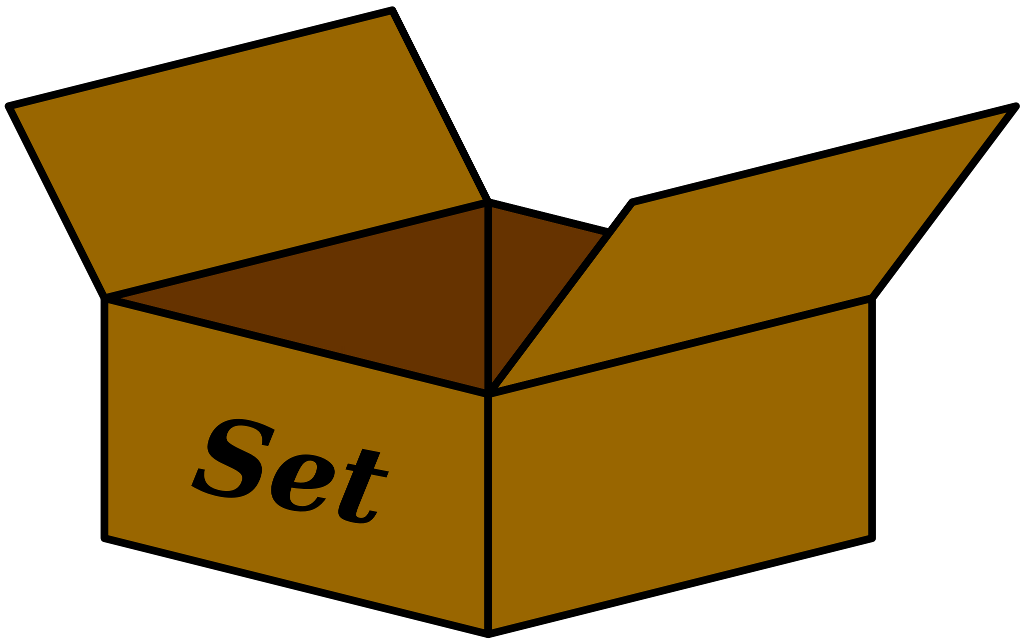
A ***set*** is an unordered collection of distinct objects, which may be anything, including other sets.



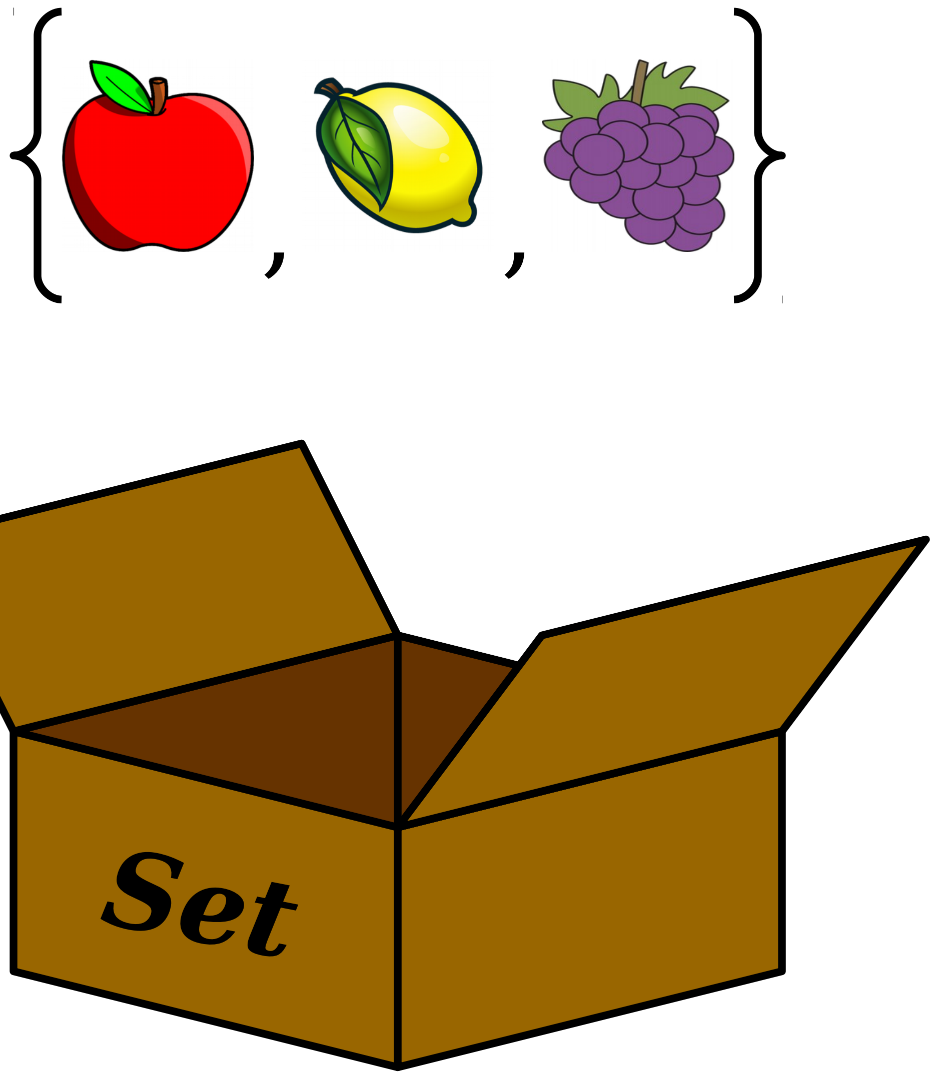
A ***set*** is an unordered collection of distinct objects, which may be anything, including other sets.



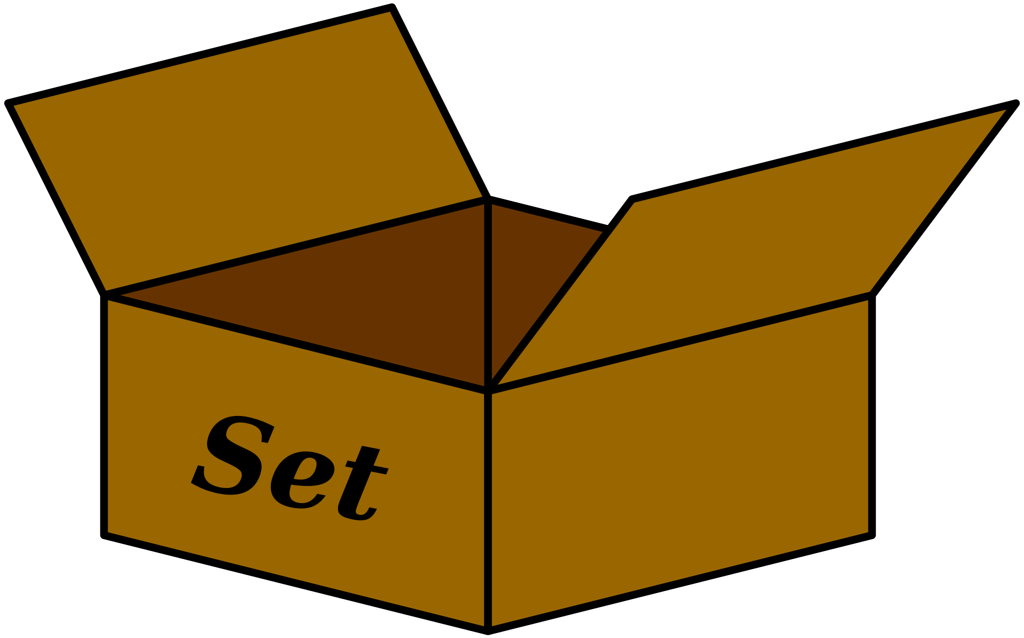
Two sets are equal when they have the same contents, ignoring order.



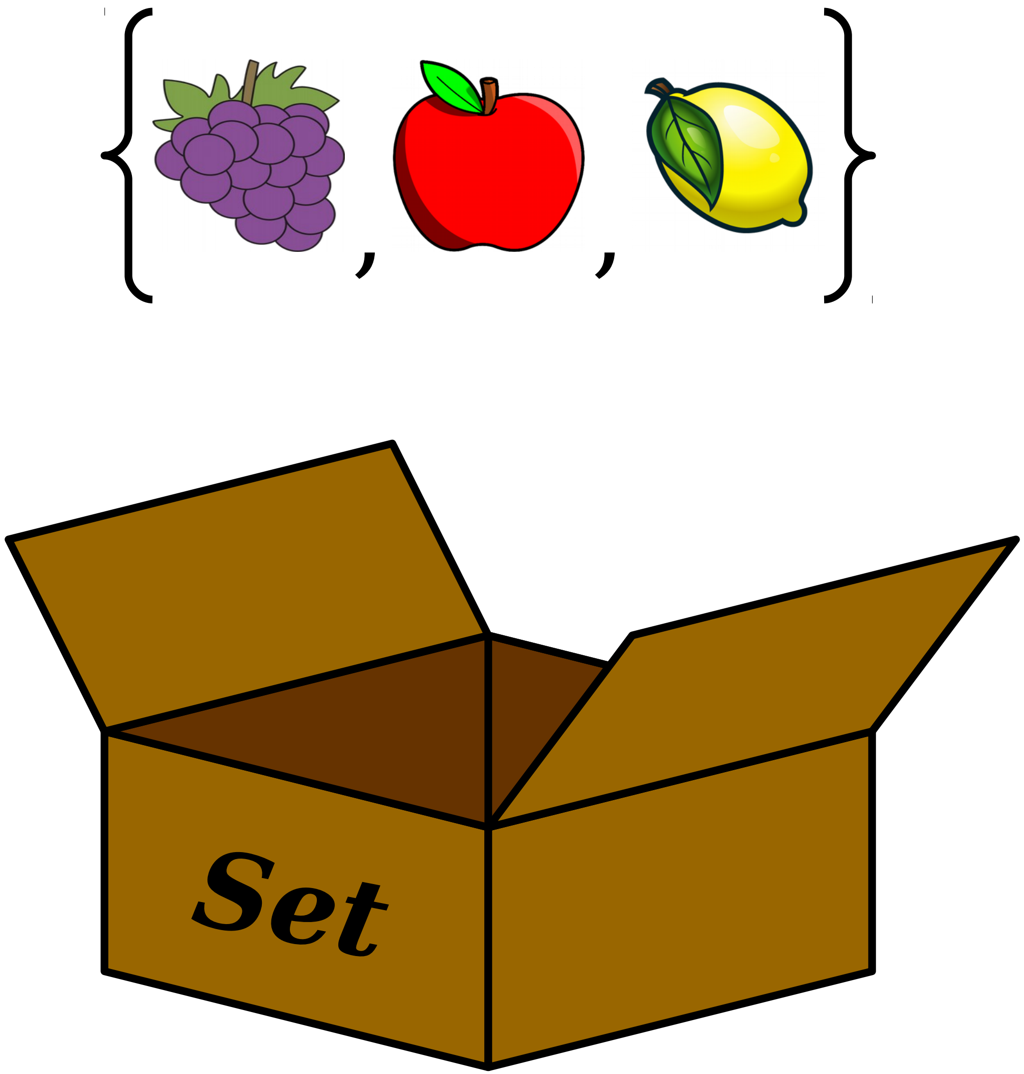
Two sets are equal when they have the same contents, ignoring order.



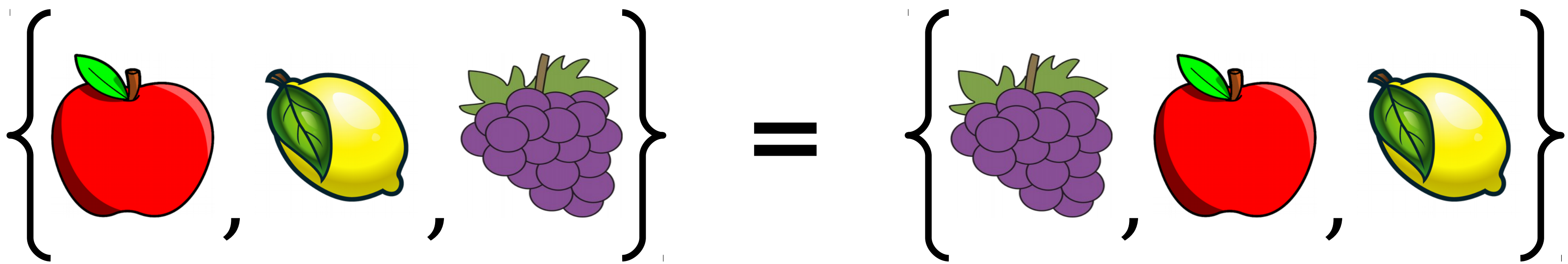
Two sets are equal when they have the same contents, ignoring order.



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

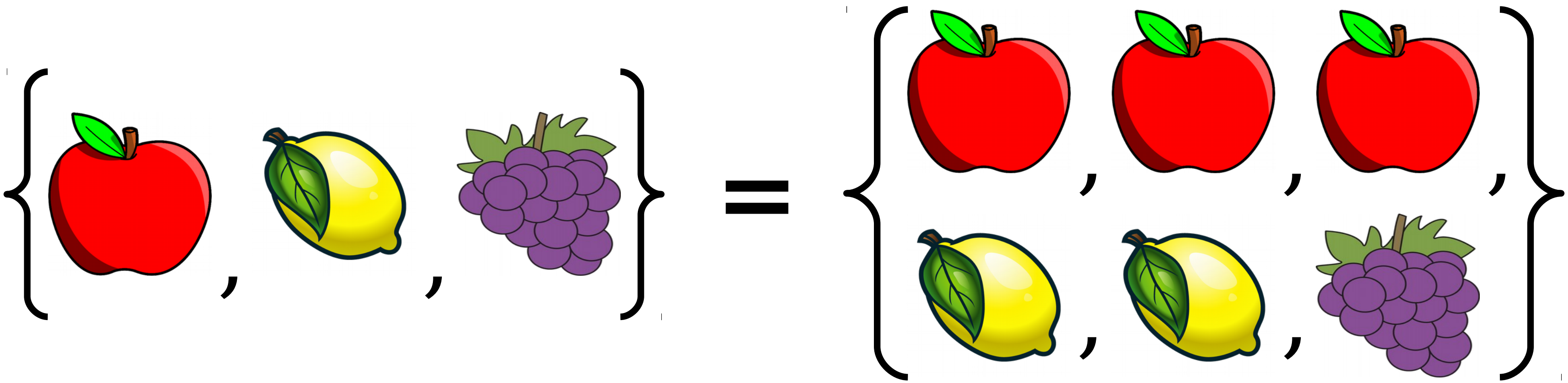


Two sets are equal when they have the same contents, ignoring order.



These are two different descriptions of exactly the same set.

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

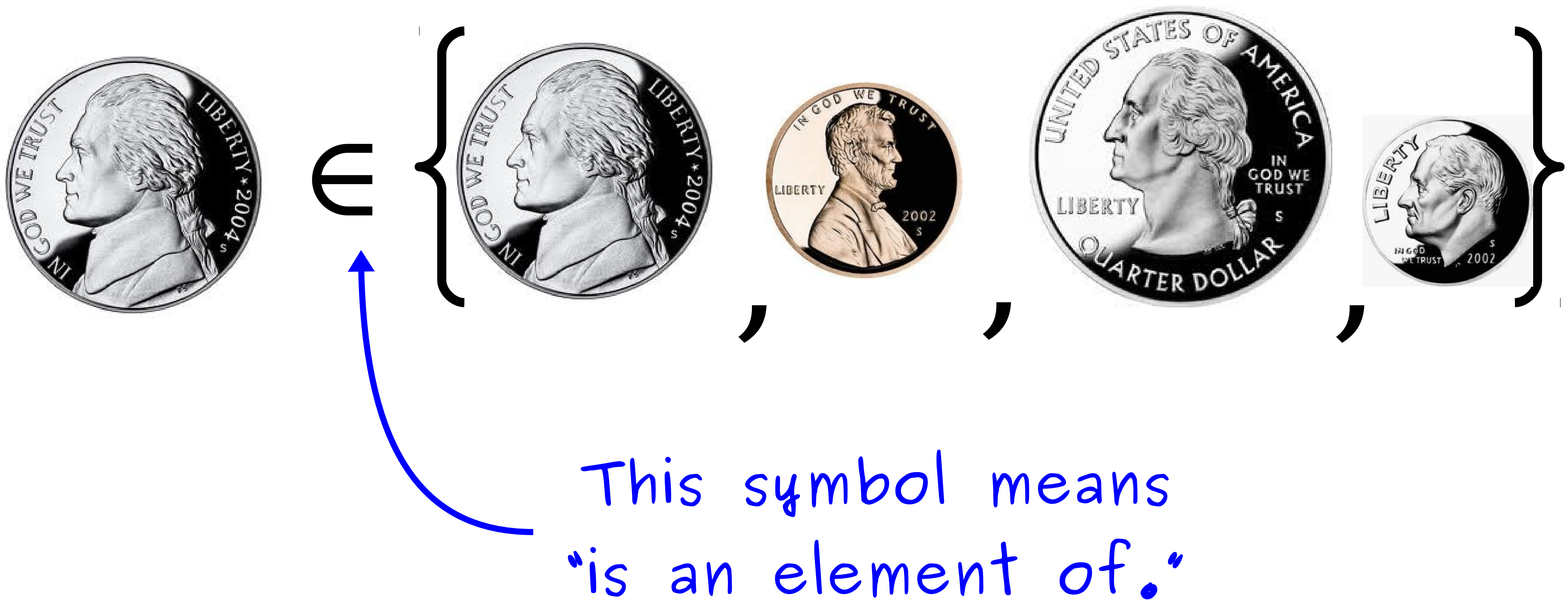


These are two different descriptions of exactly the same set.

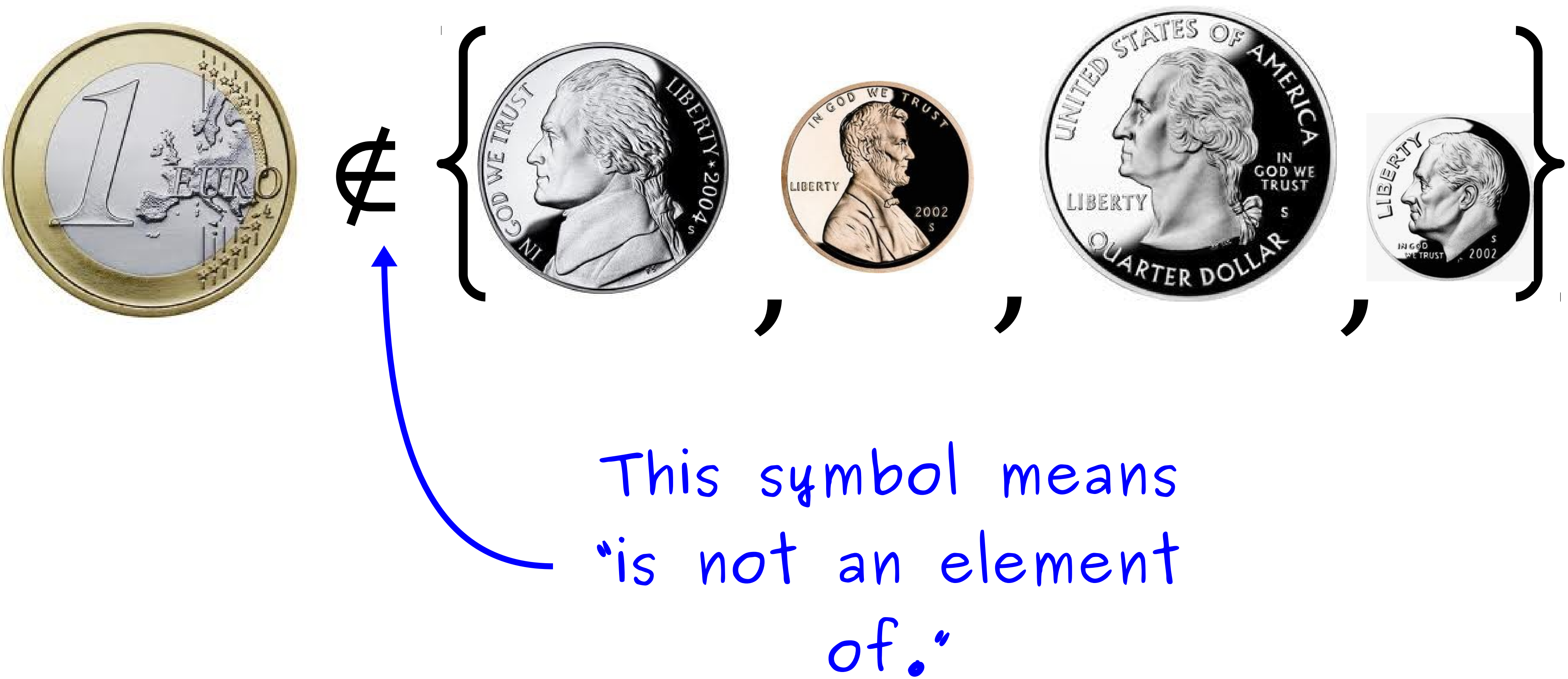
Sets cannot contain duplicate elements.
Any repeated elements are ignored.

The objects that make up a set are called the ***elements*** of that set.

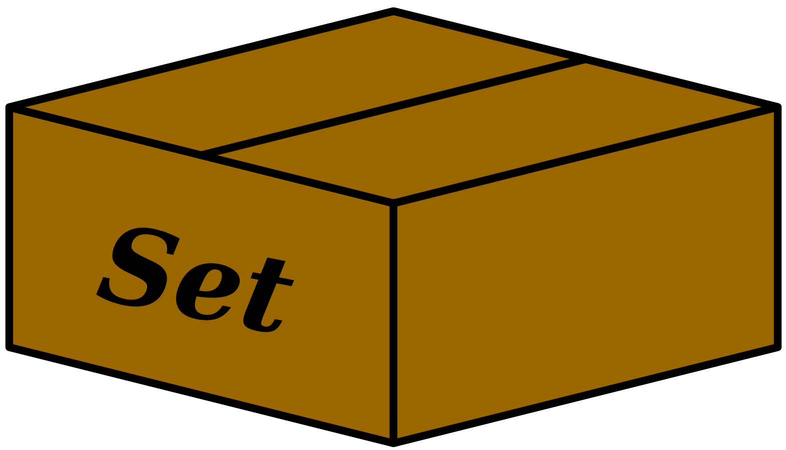
The objects that make up a set are called the ***elements*** of that set.



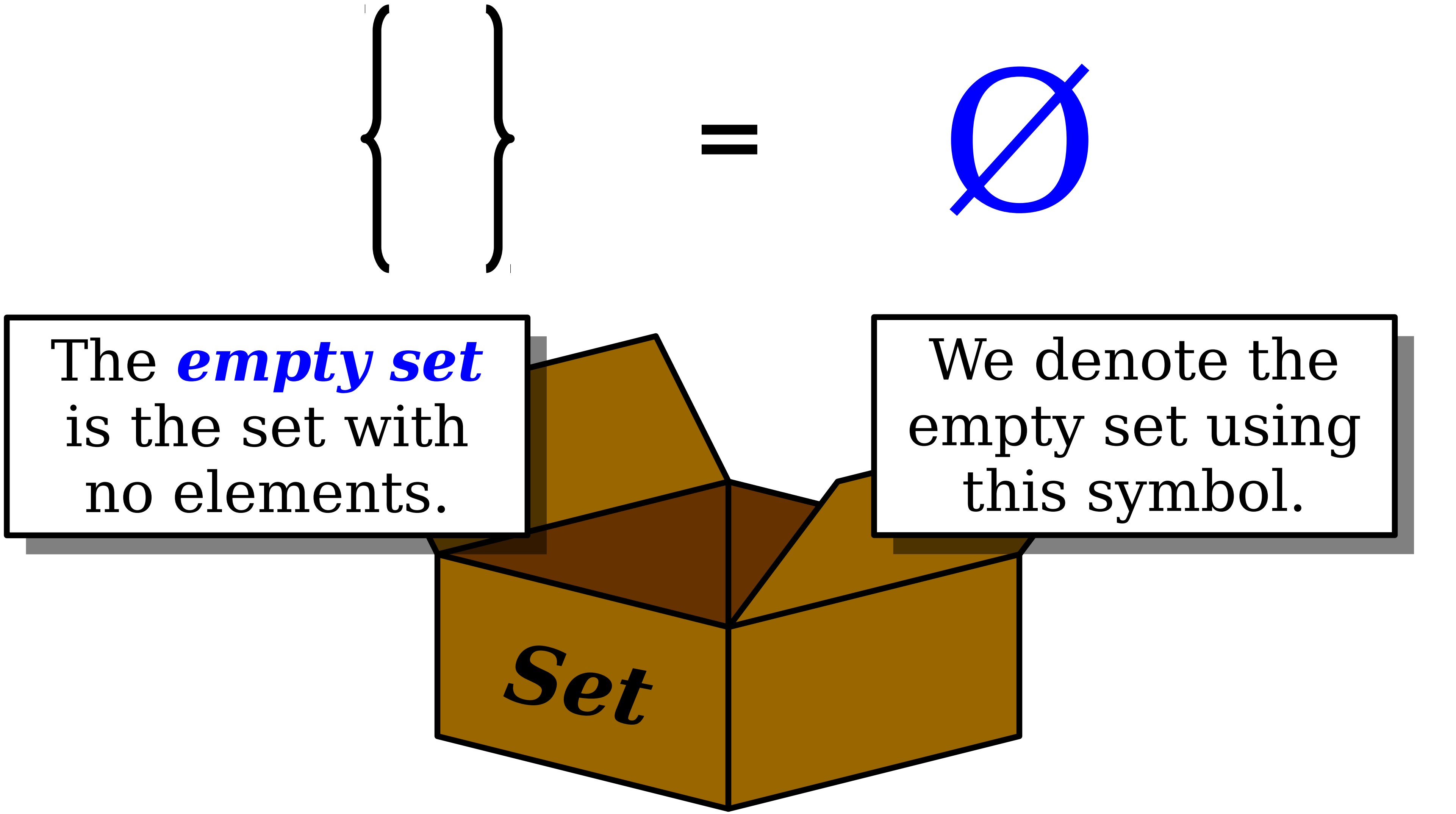
The objects that make up a set are called the ***elements*** of that set.



The objects that make up a set are called the ***elements*** of that set.



Sets can contain any number of elements.

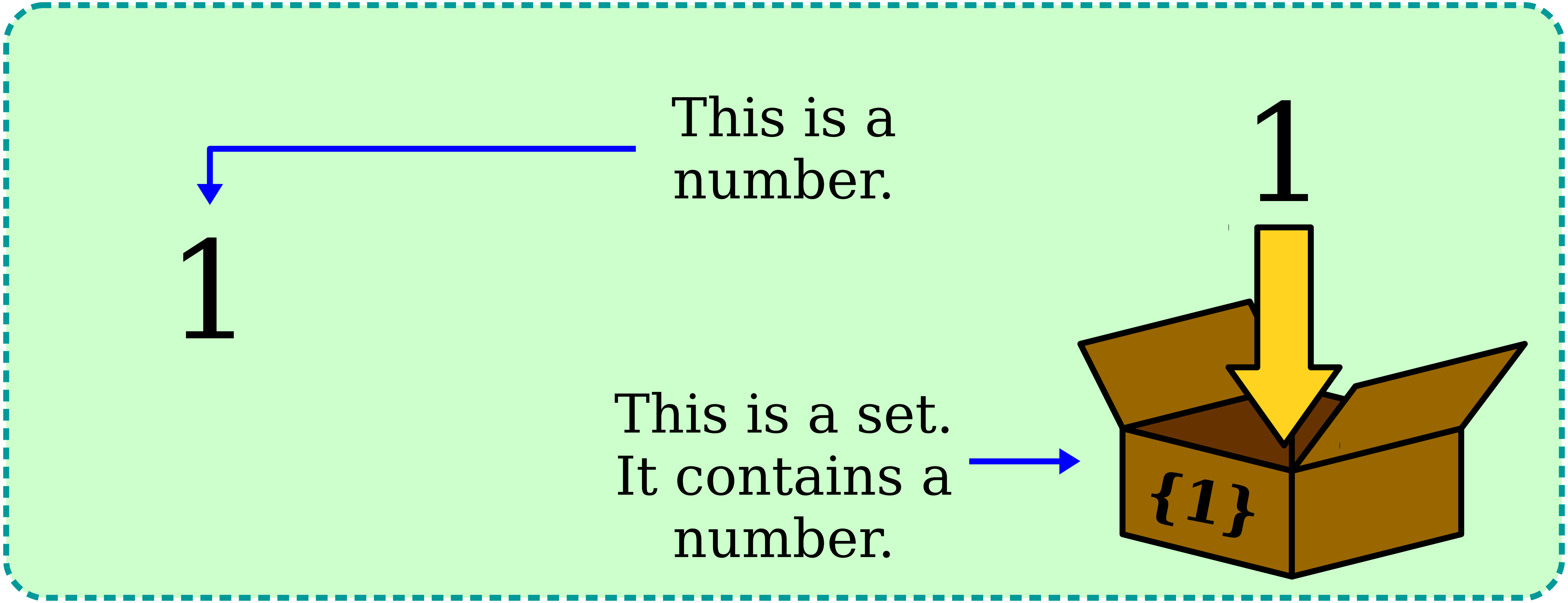


Sets can contain any number of elements.

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

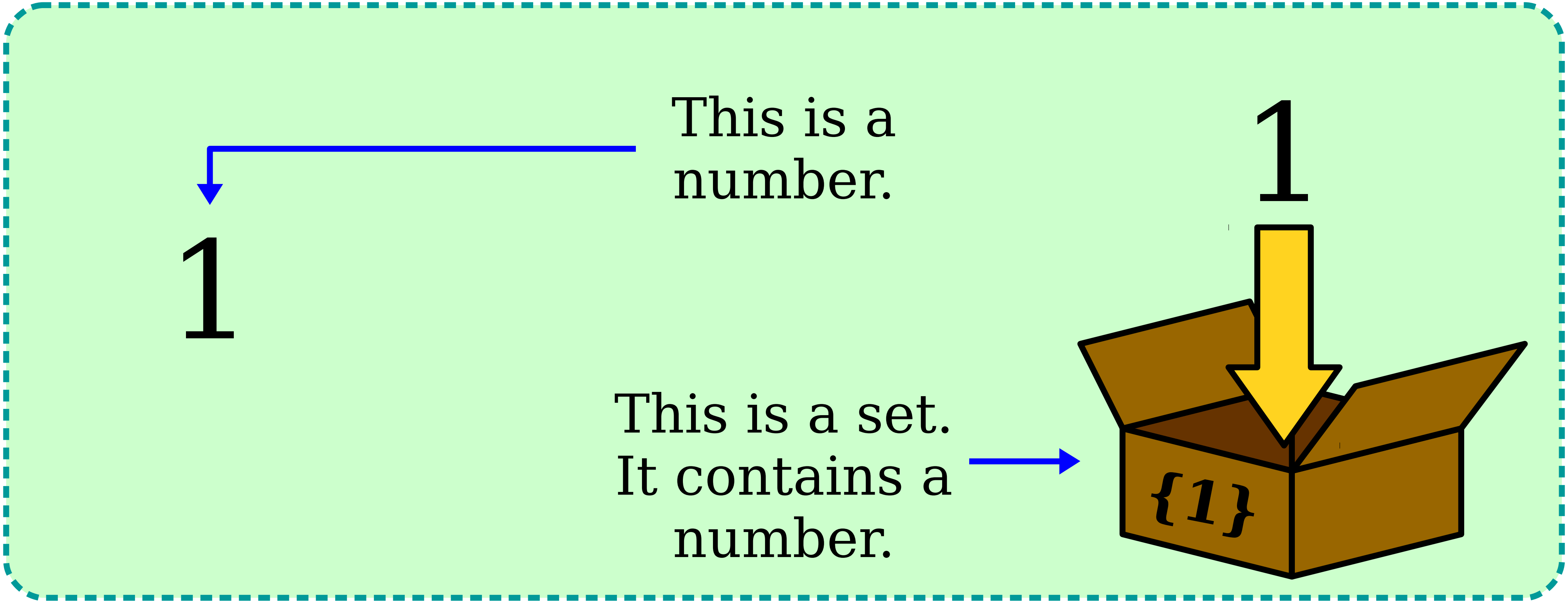
Question: Are these objects equal?

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



Question: Are these objects equal?

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

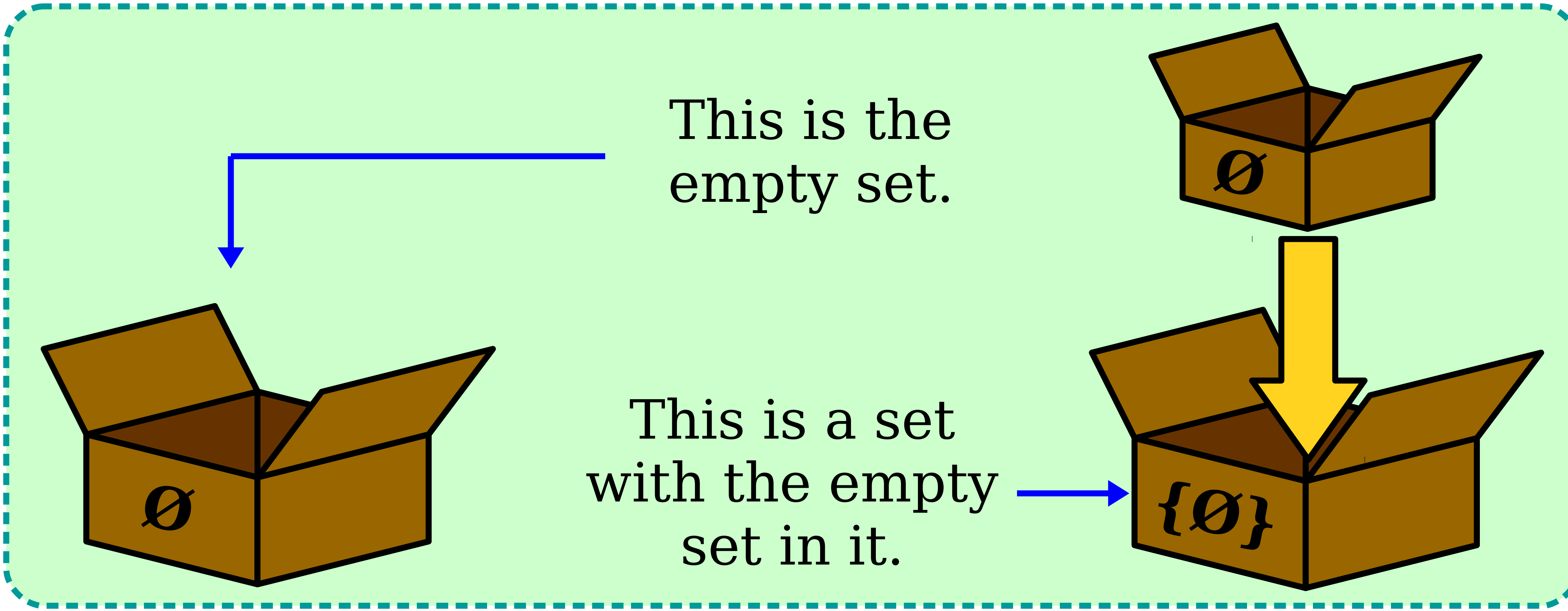


Question: Are these objects equal?

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

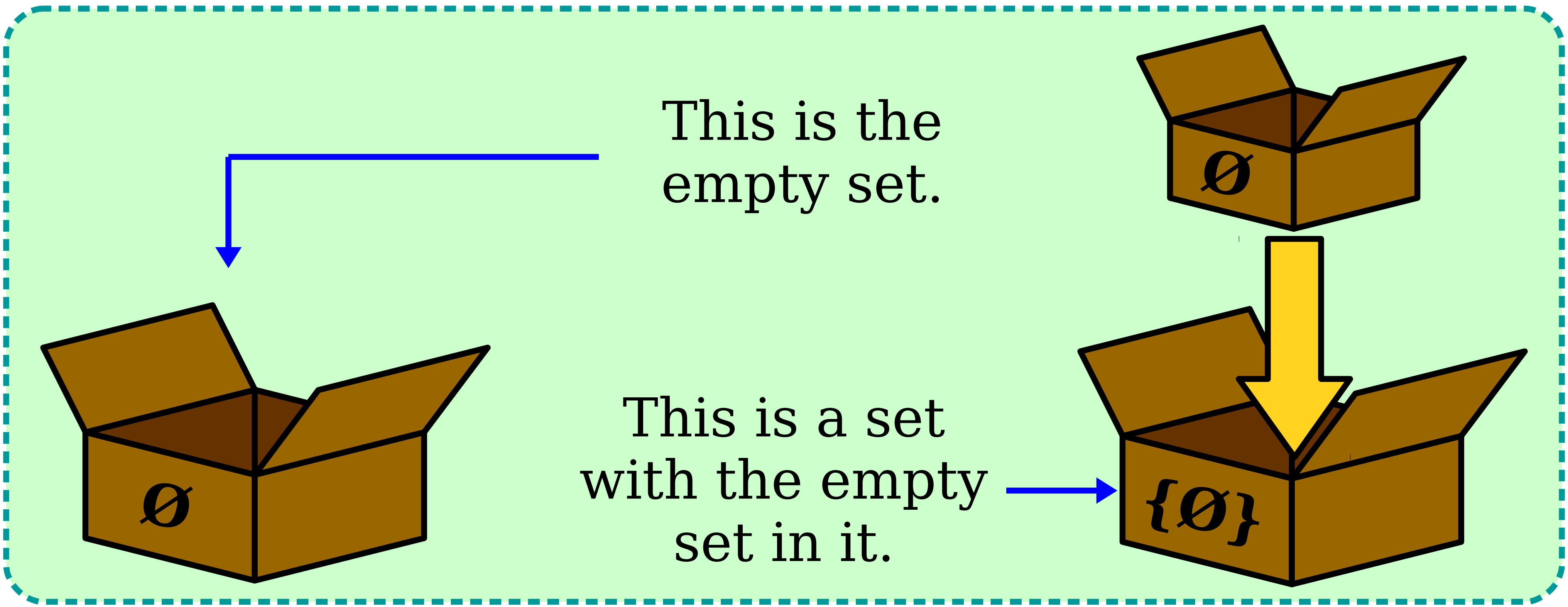
Question: Are these objects equal?

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



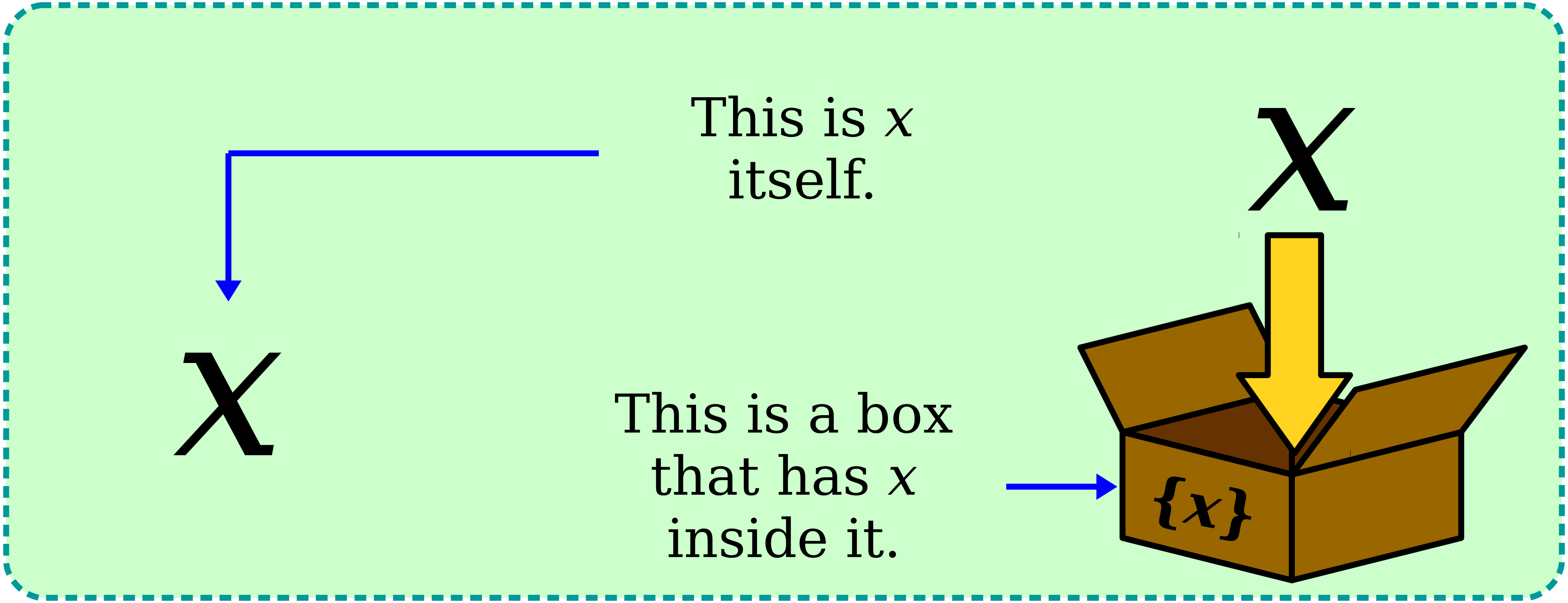
Question: Are these objects equal?

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



Question: Are these objects equal?

$$x \neq \{x\}$$



No object *x* is equal to the set containing *x*.

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***.
  - $e \in \mathbb{R}$, $\pi \in \mathbb{R}$, $4 \in \mathbb{R}$, -137, 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

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

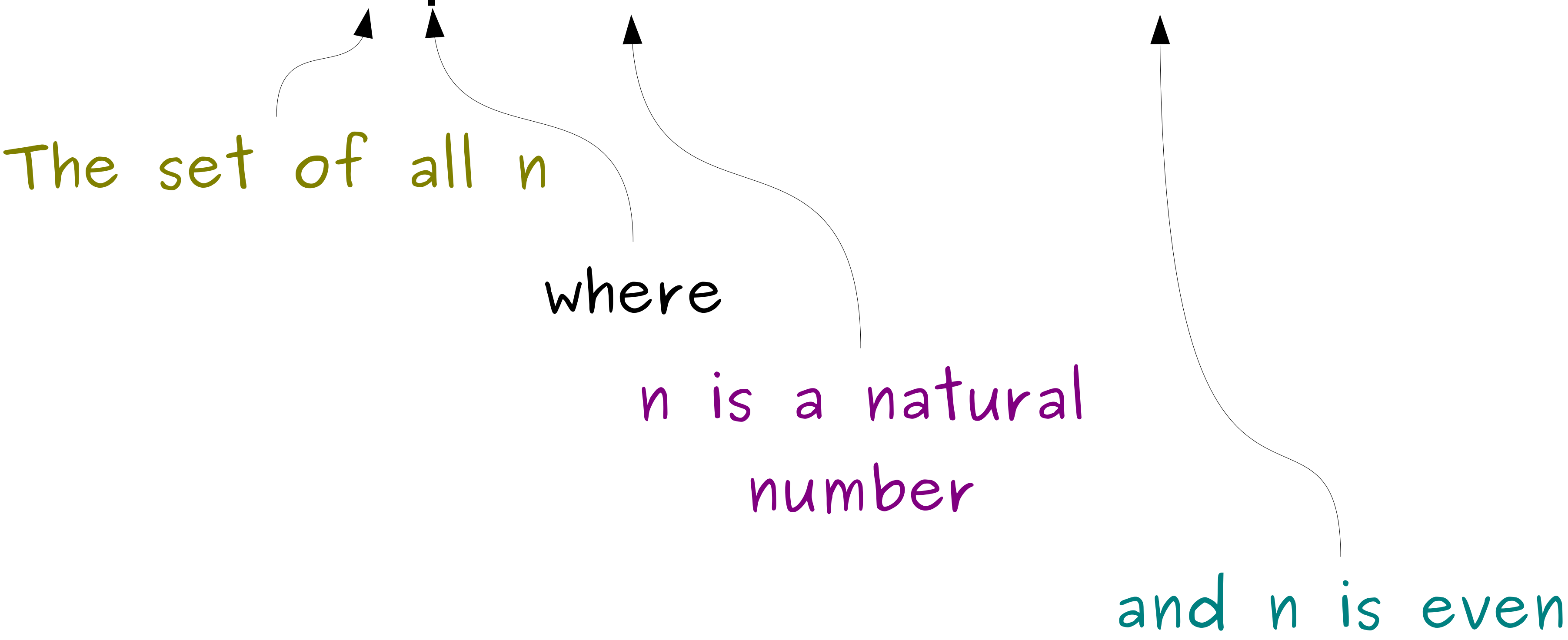
The set of all n

where

n is a natural number

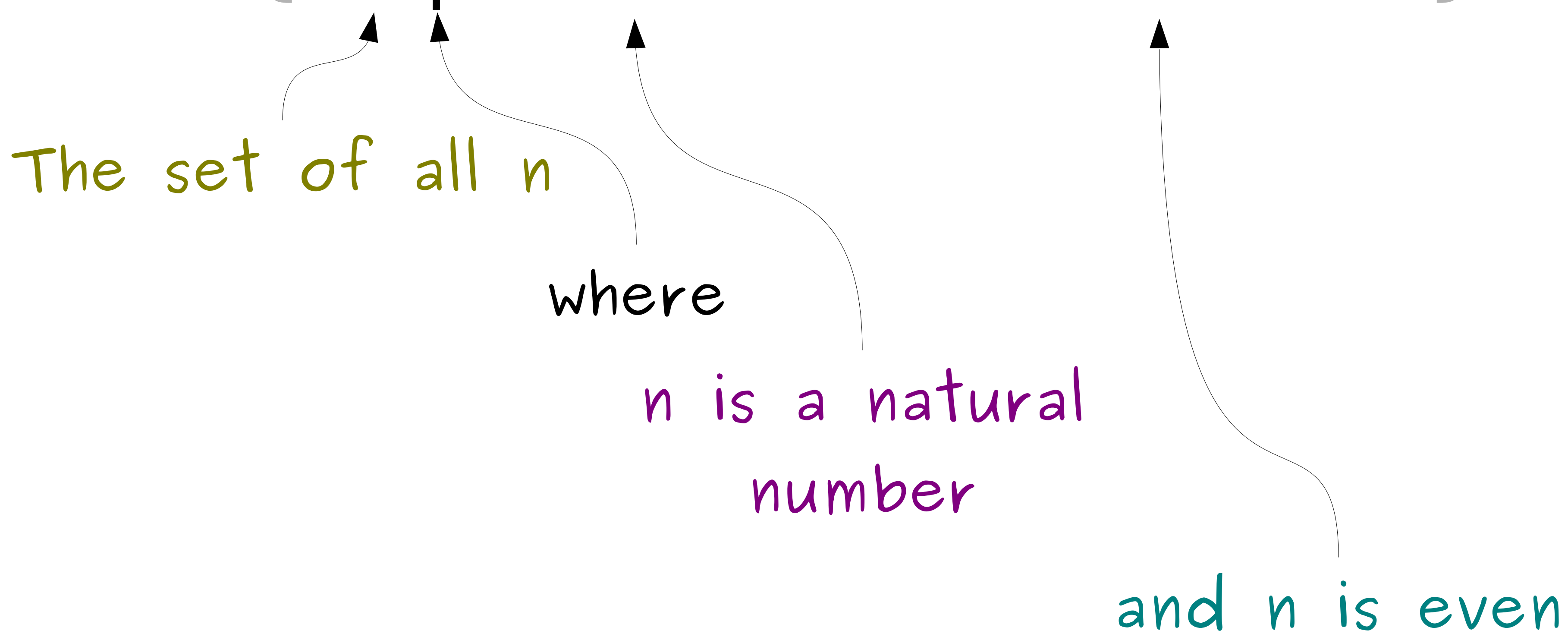
Even Natural Numbers

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



Even Natural Numbers

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



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

Set Builder Notation

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

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

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

- For example:

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

$\{\ C \mid C \text{ is a US coin}\ \}$

$\{\ n \in \mathbb{N} \mid n < 3\ \}$ **(the set {0, 1, 2})**

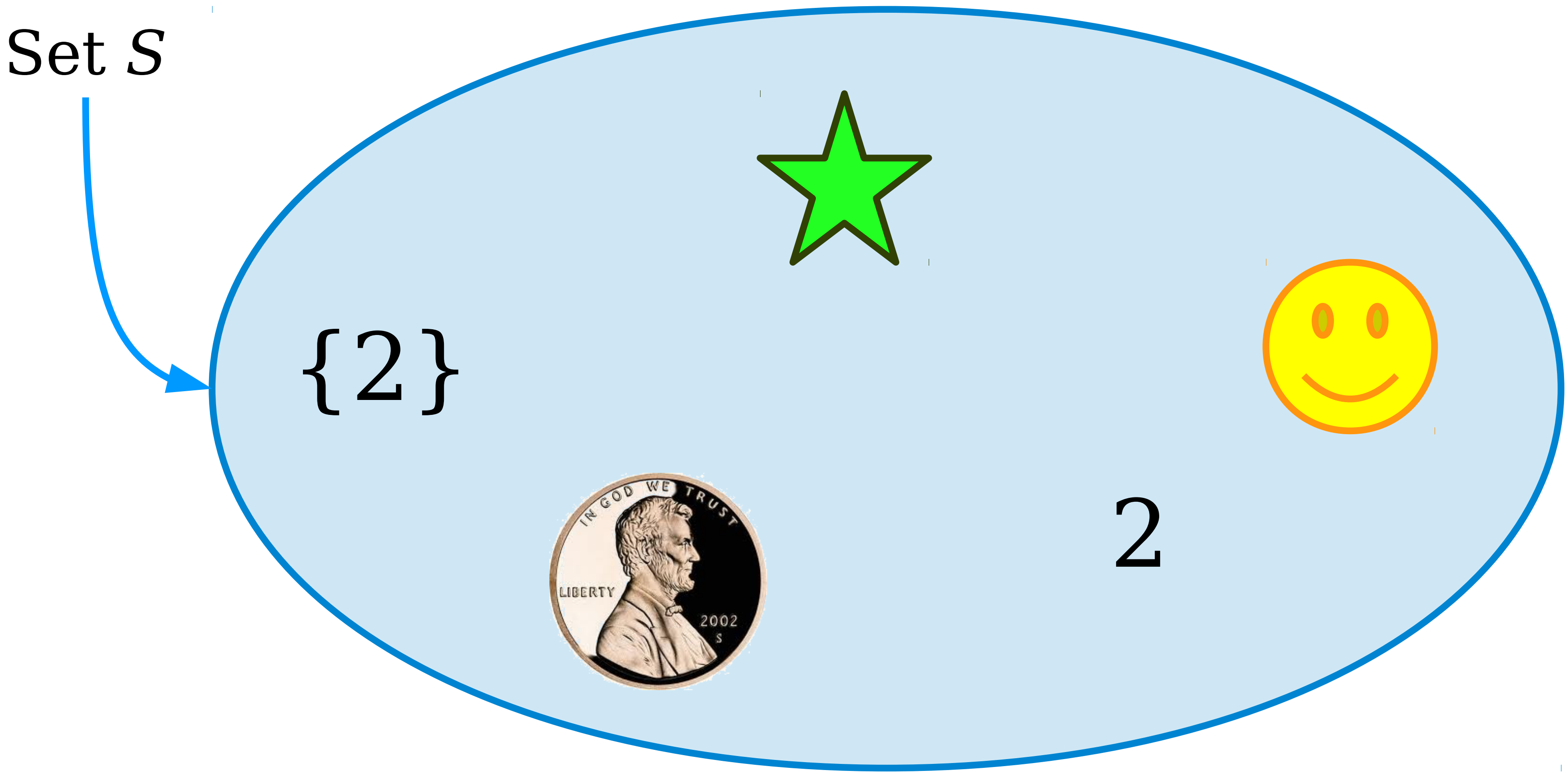
$\{\ r \in \mathbb{R} \mid r < 3\ \}$

Subsets and Power Sets

Subsets

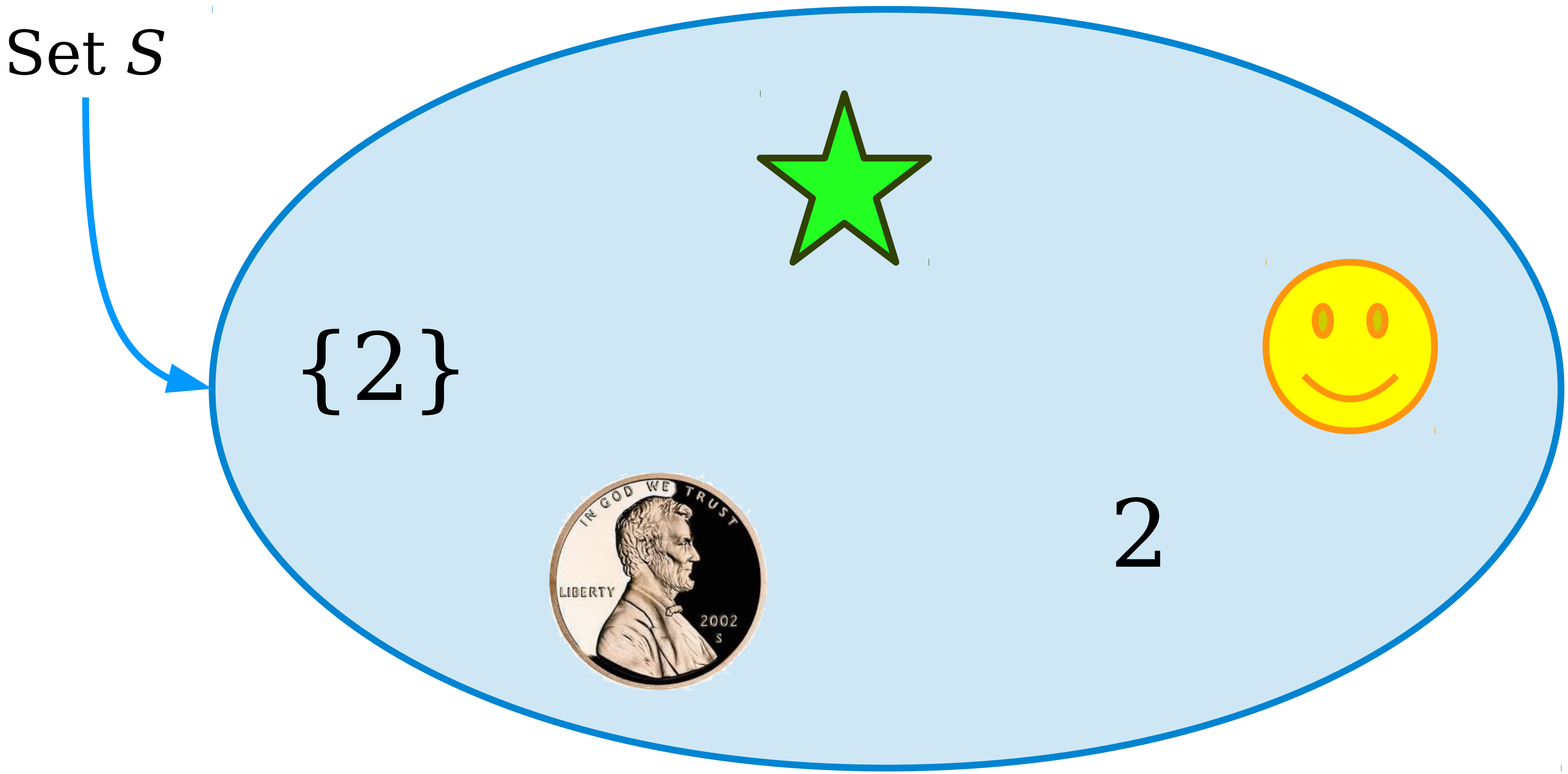
- A set S is called a ***subset*** of a set T (denoted $\boldsymbol{S \subseteq T}$) if all elements of S are also elements of T.
- Examples:
  - $\{ 1, 2, 3 \} \subseteq \{ 1, 2, 3, 4 \}$
  - $\{ b, c \} \subseteq \{ a, b, c, d \}$
  - { H, He, Li } ⊆ { H, He, Li }
  - $\mathbb{N} \subseteq \mathbb{Z}$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

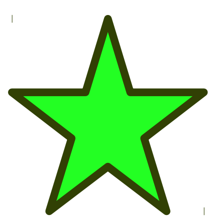
Subsets and Elements



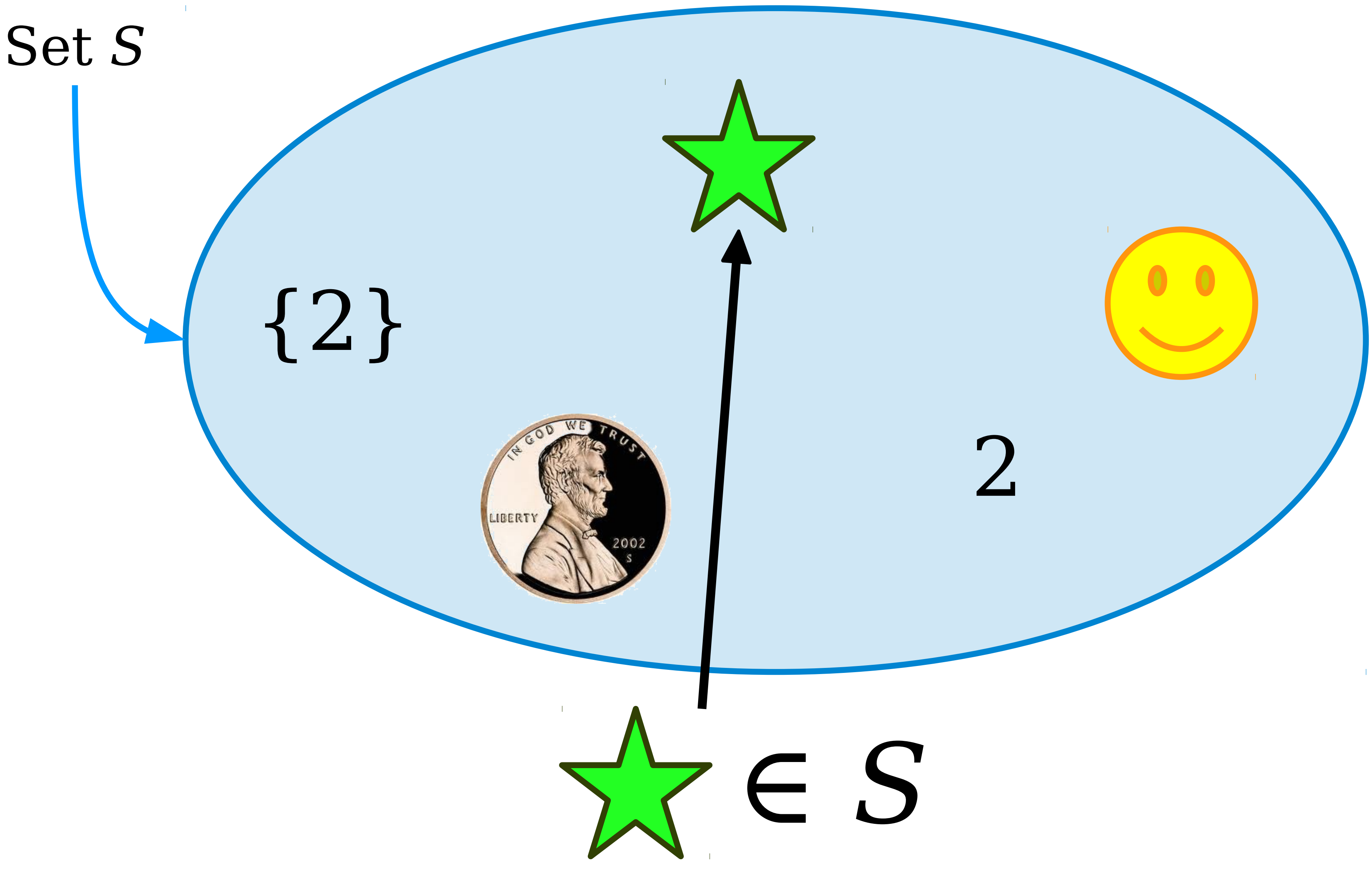
$S = \{$ 2, , {2}, ,  $\}$

Subsets and Elements

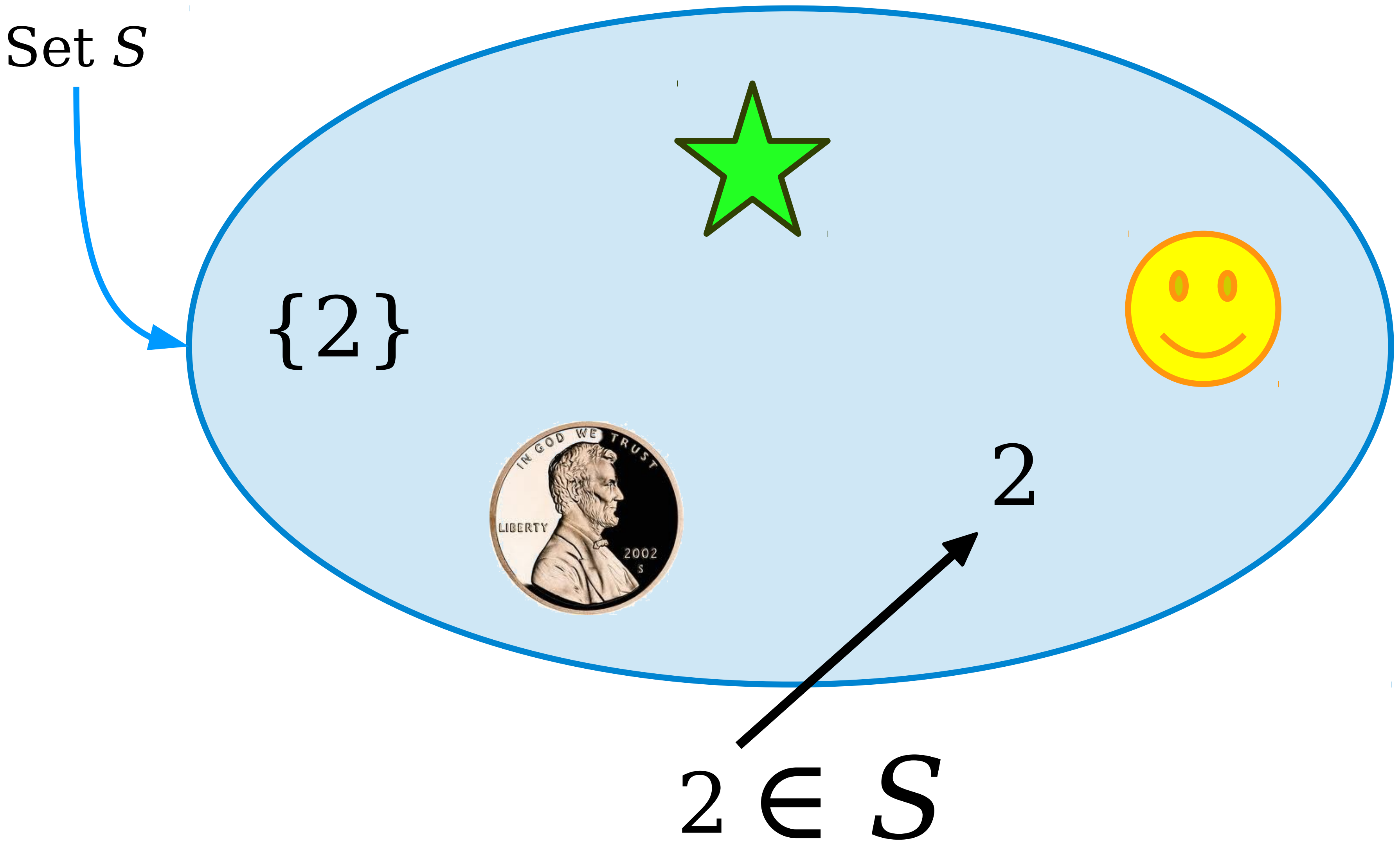


 $\in S$

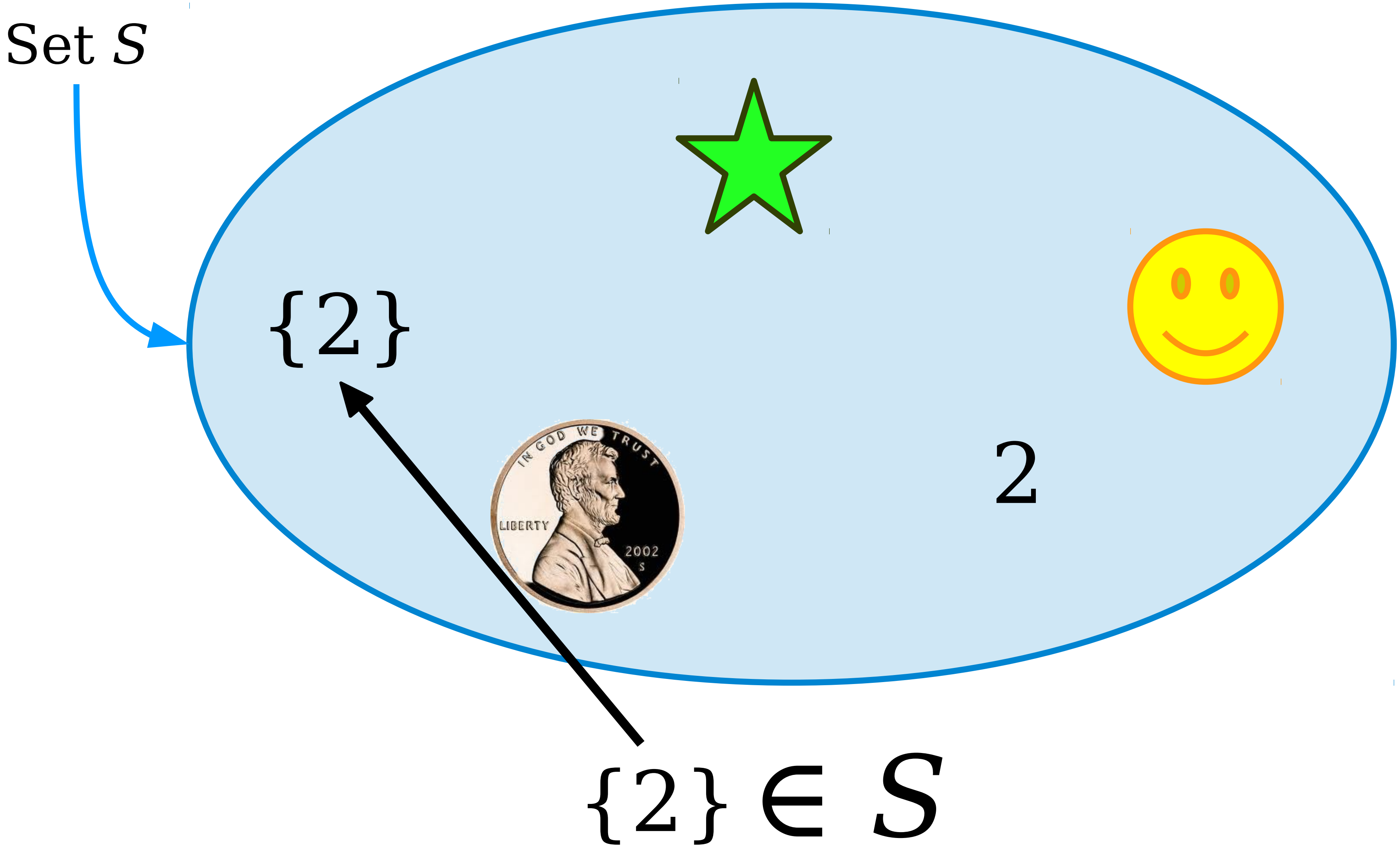
Subsets and Elements



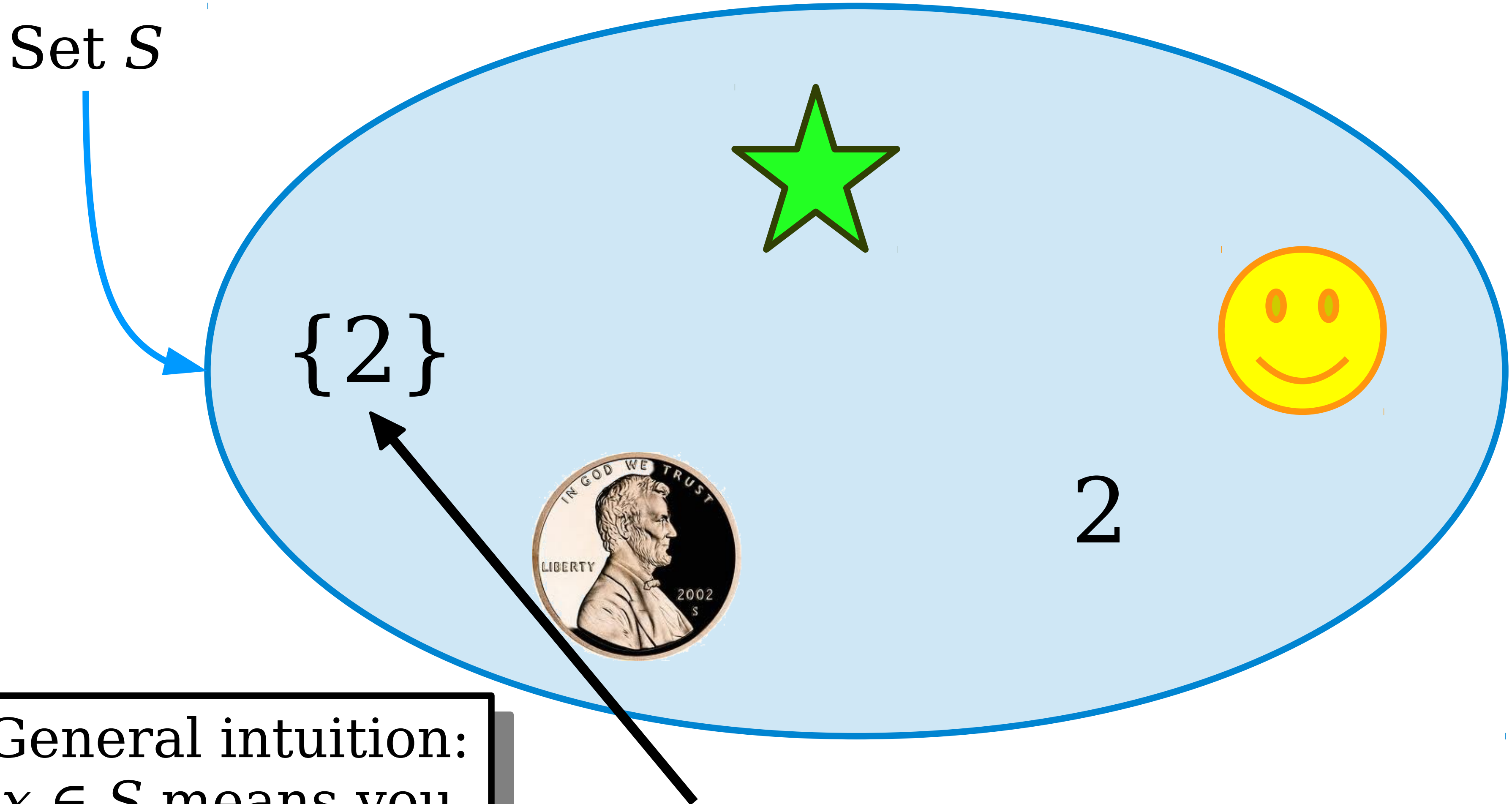
Subsets and Elements



Subsets and Elements



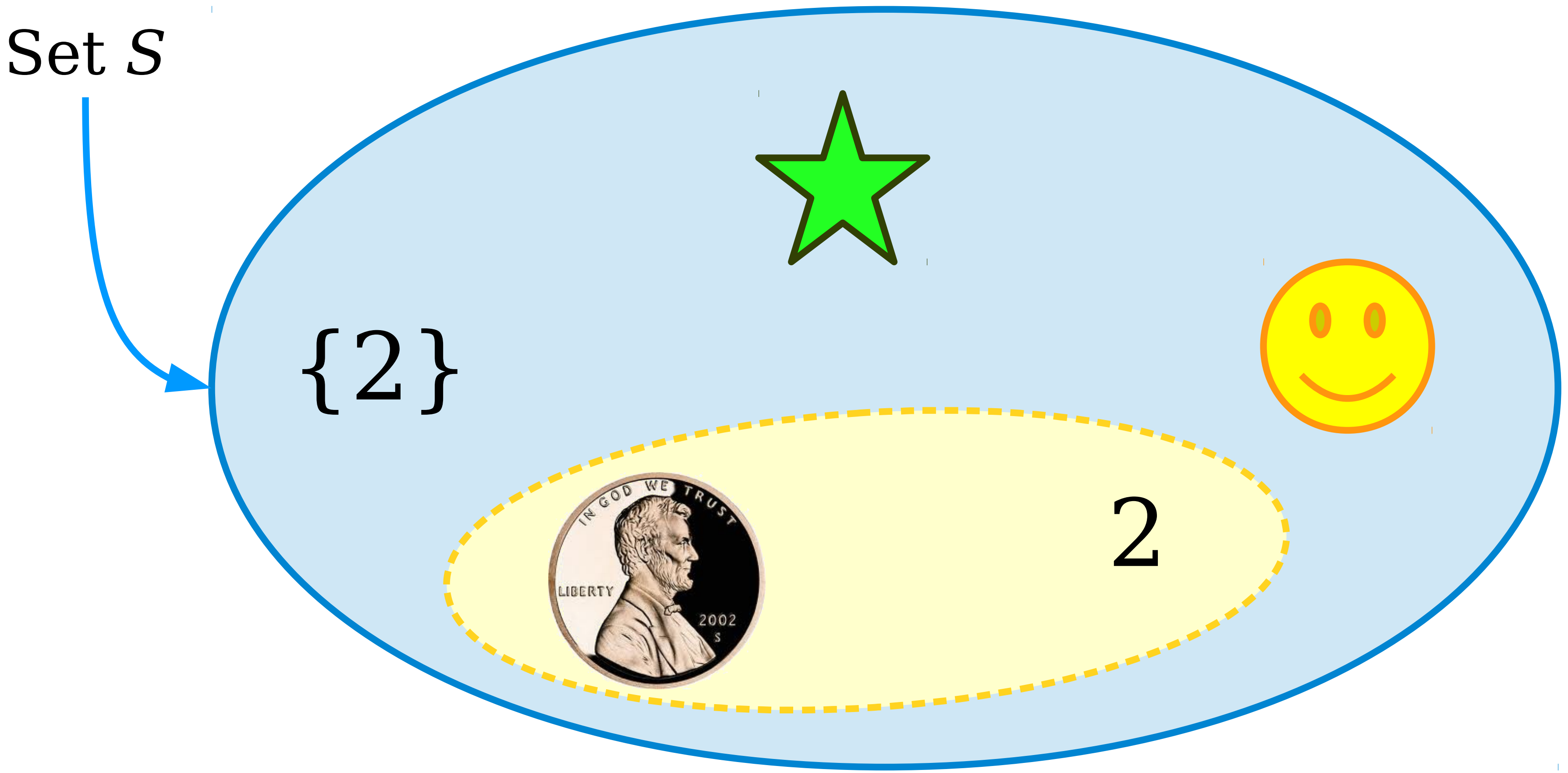
Subsets and Elements



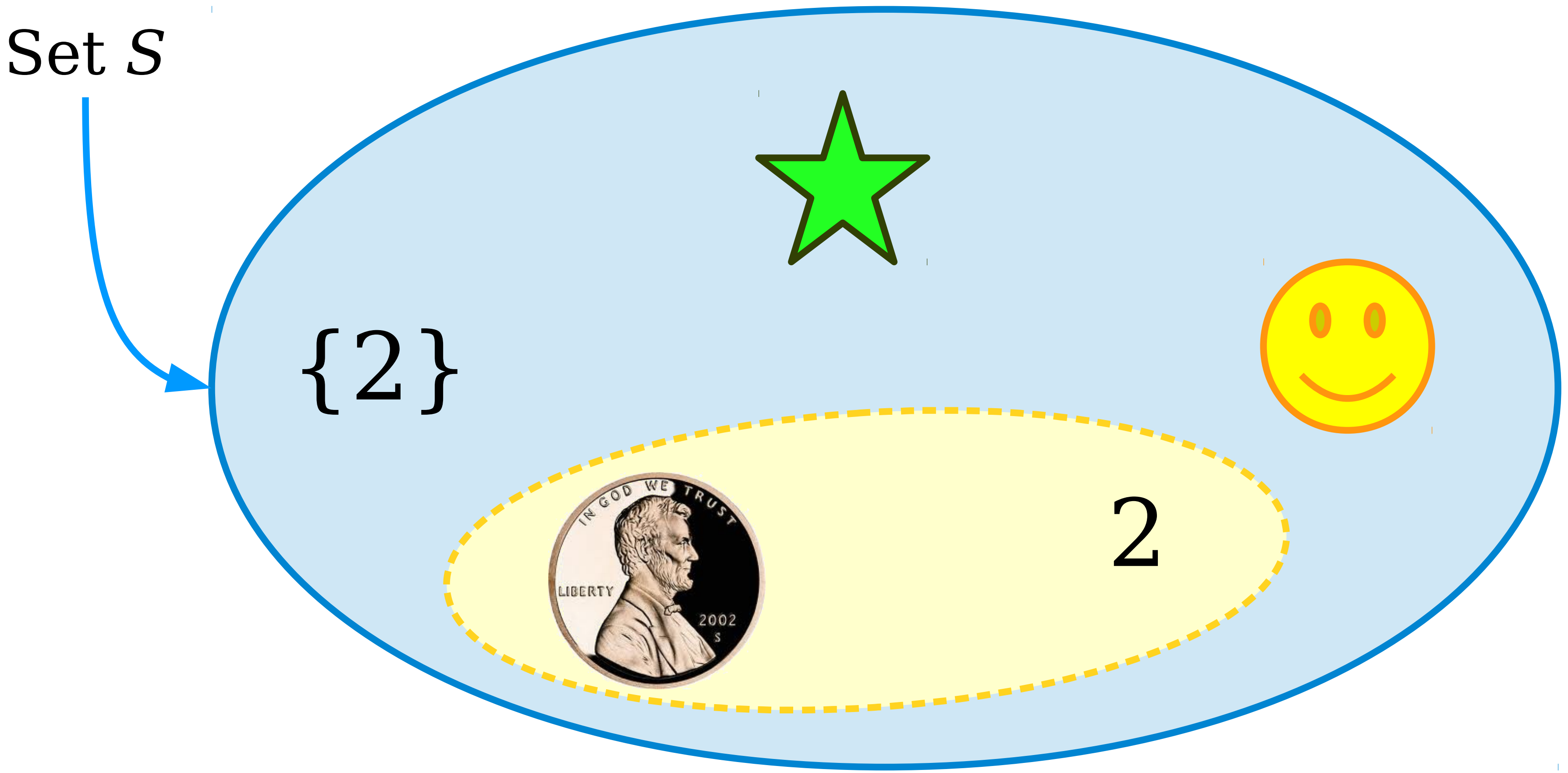
General intuition:
$x \in S$ means you
can ***point at x***
inside of S.

$\{2\} \in S$

Subsets and Elements

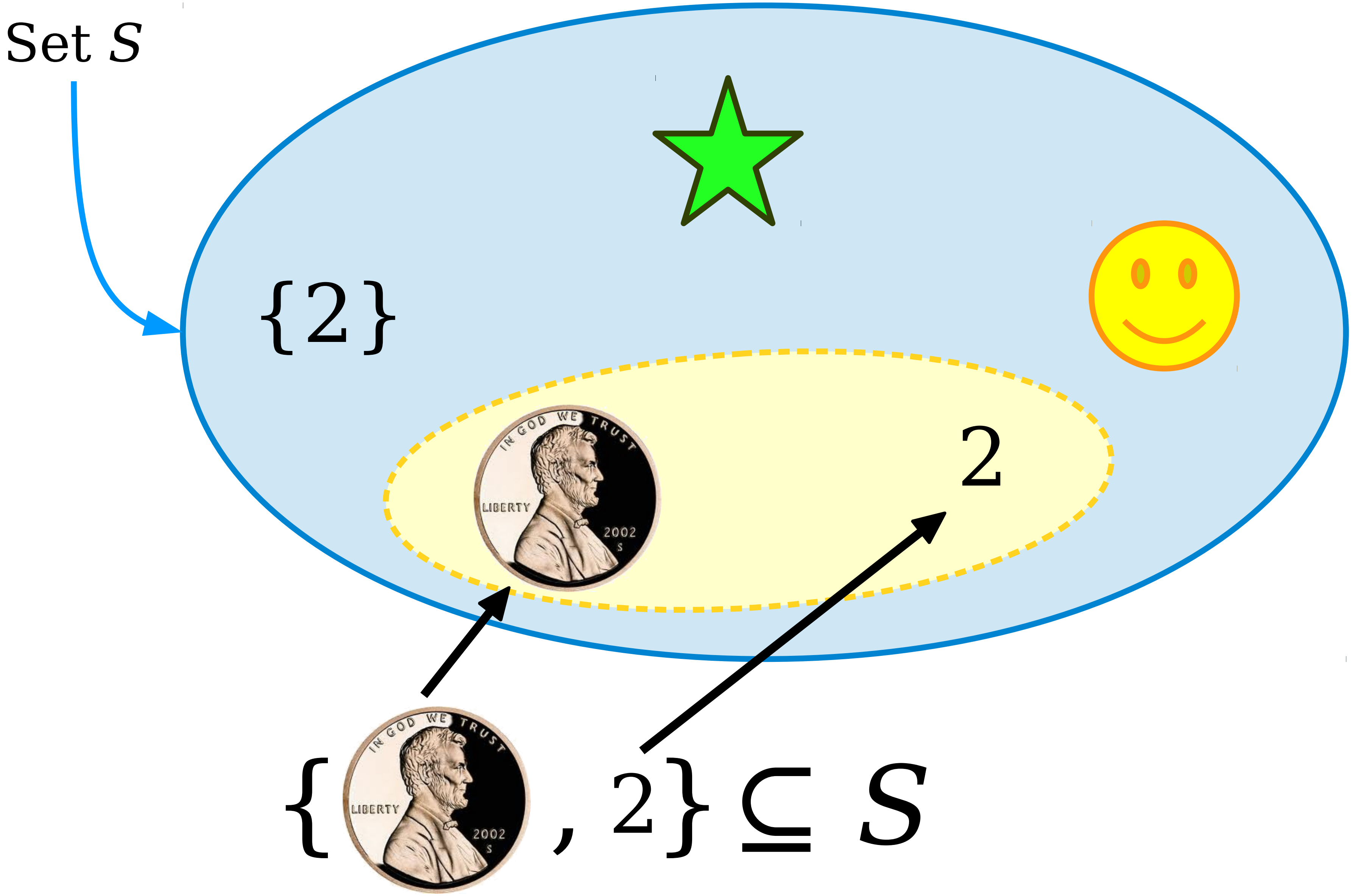


Subsets and Elements

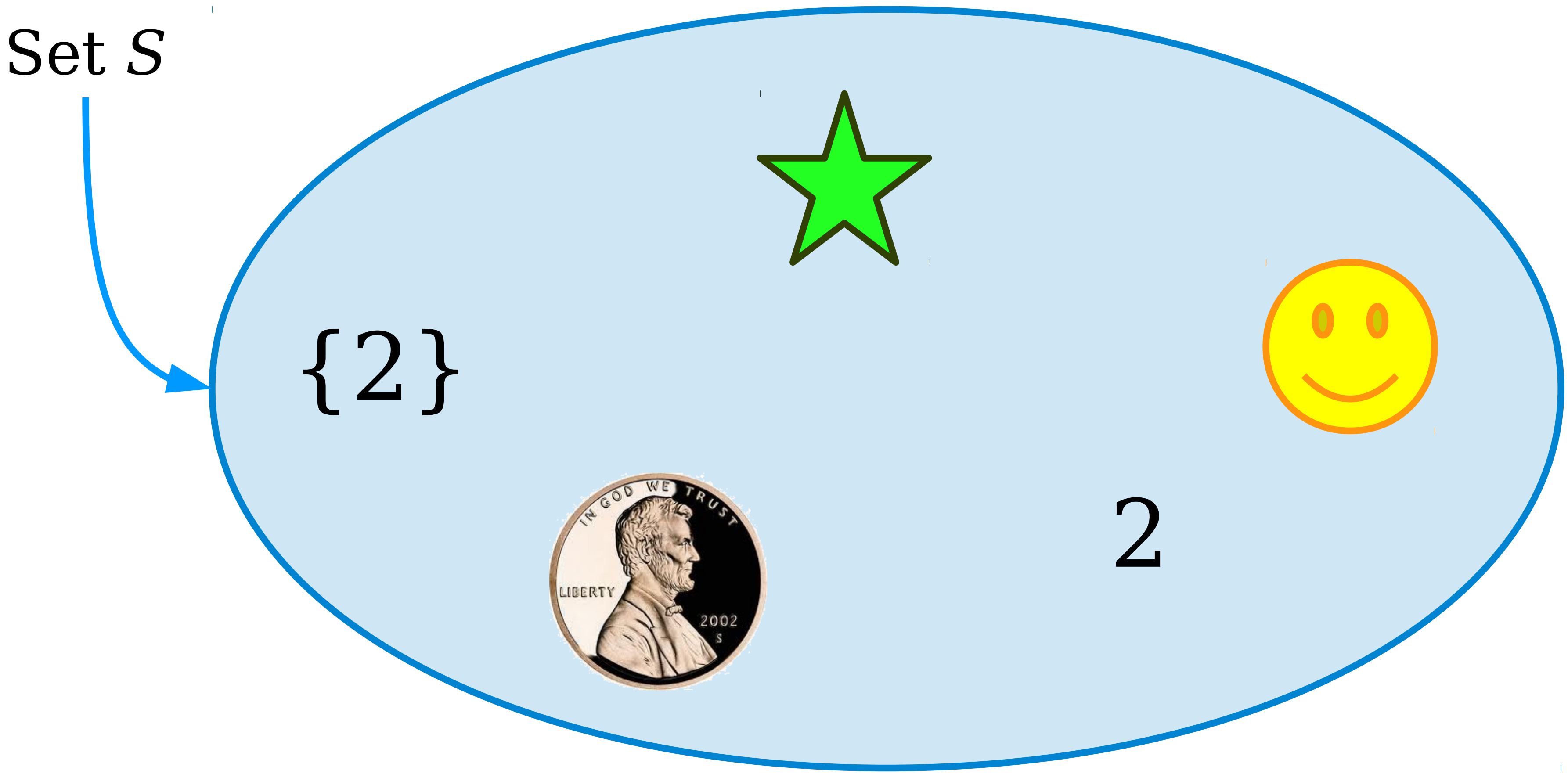


$\{$  $, 2\} \subseteq S$

Subsets and Elements

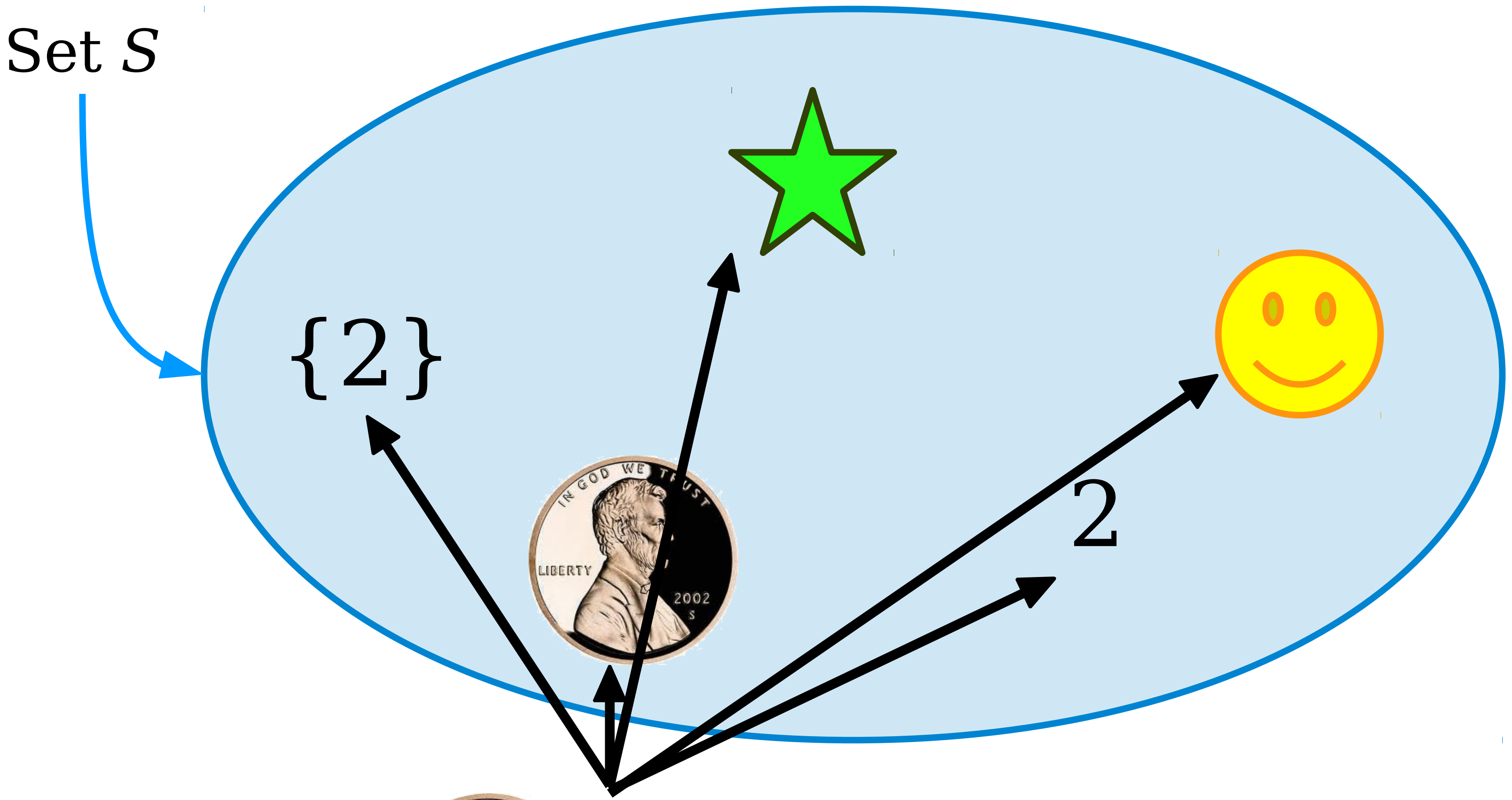


Subsets and Elements



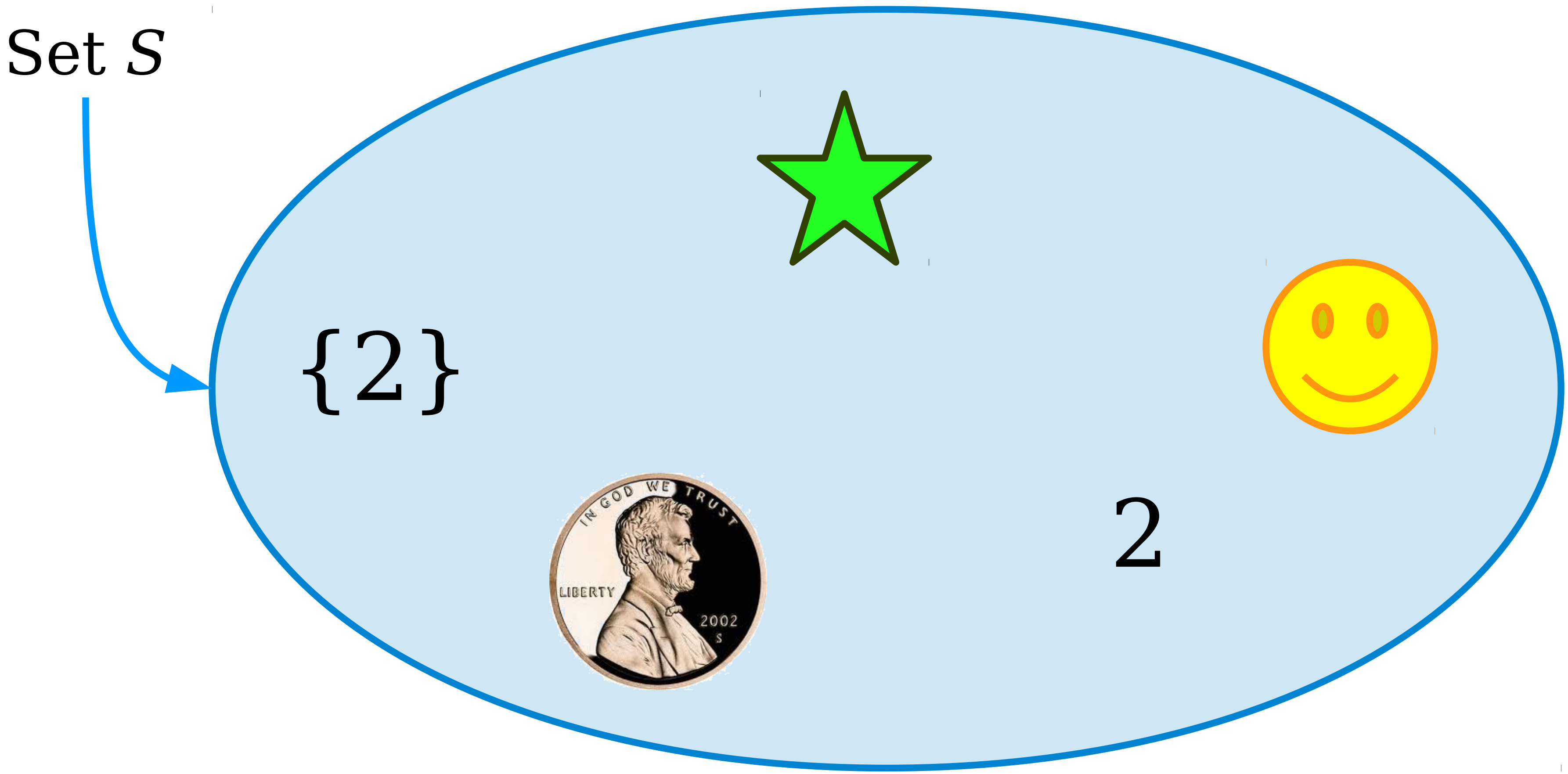
$\{$  $, 2\} \notin S$

Subsets and Elements



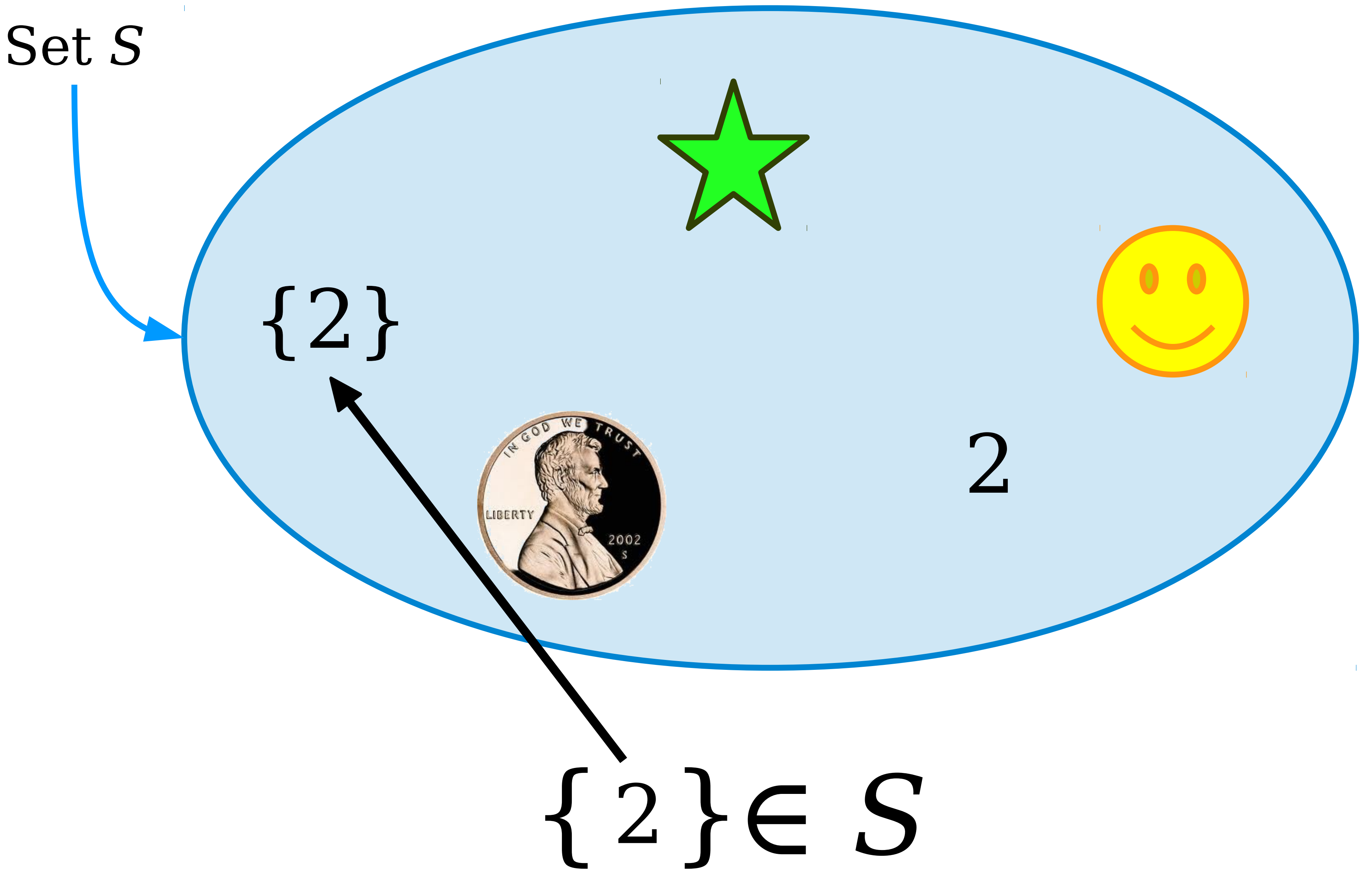
$\{$  $, 2\} \notin S$

Subsets and Elements

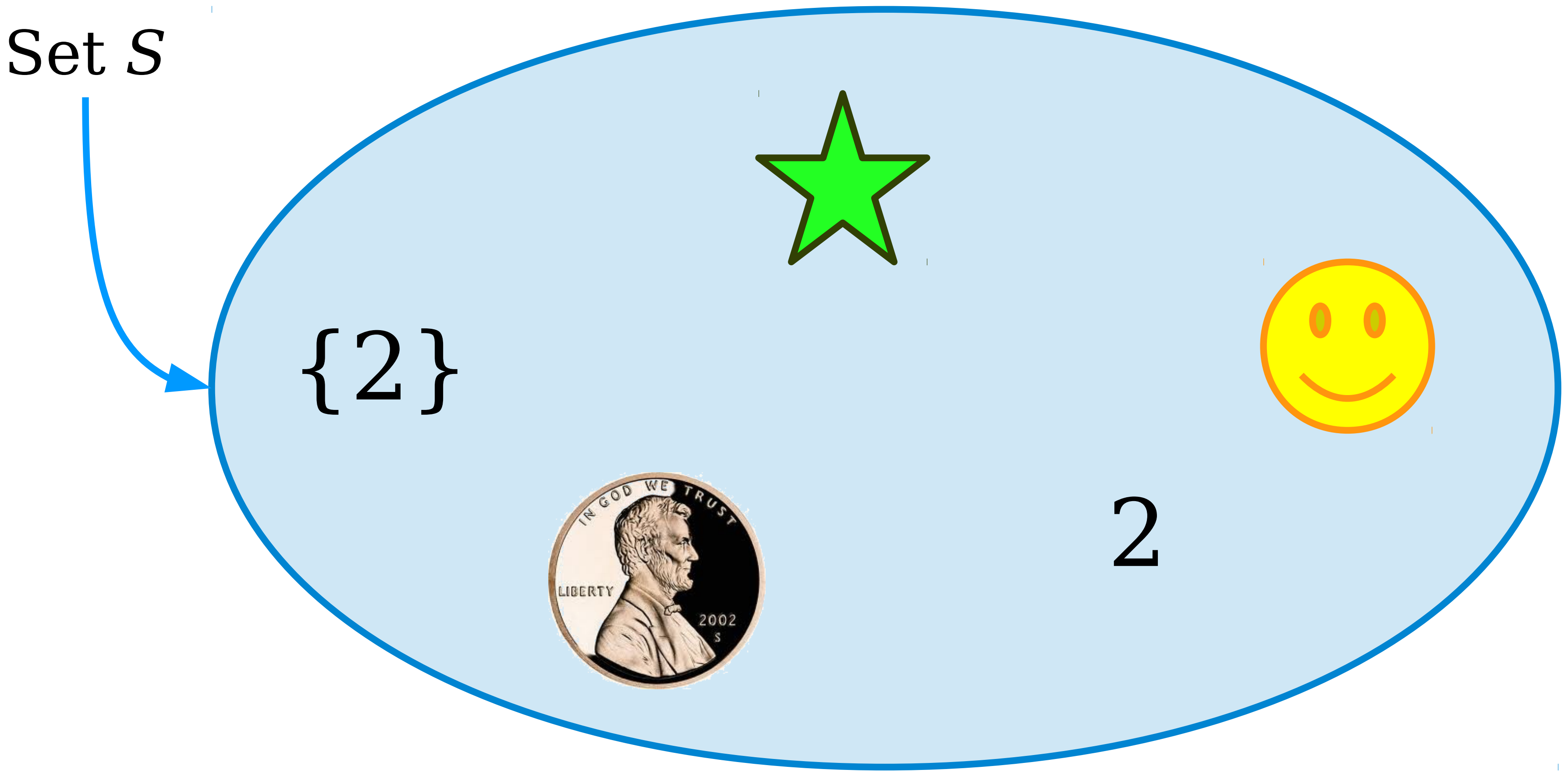


$$\{2\} \in S$$

Subsets and Elements

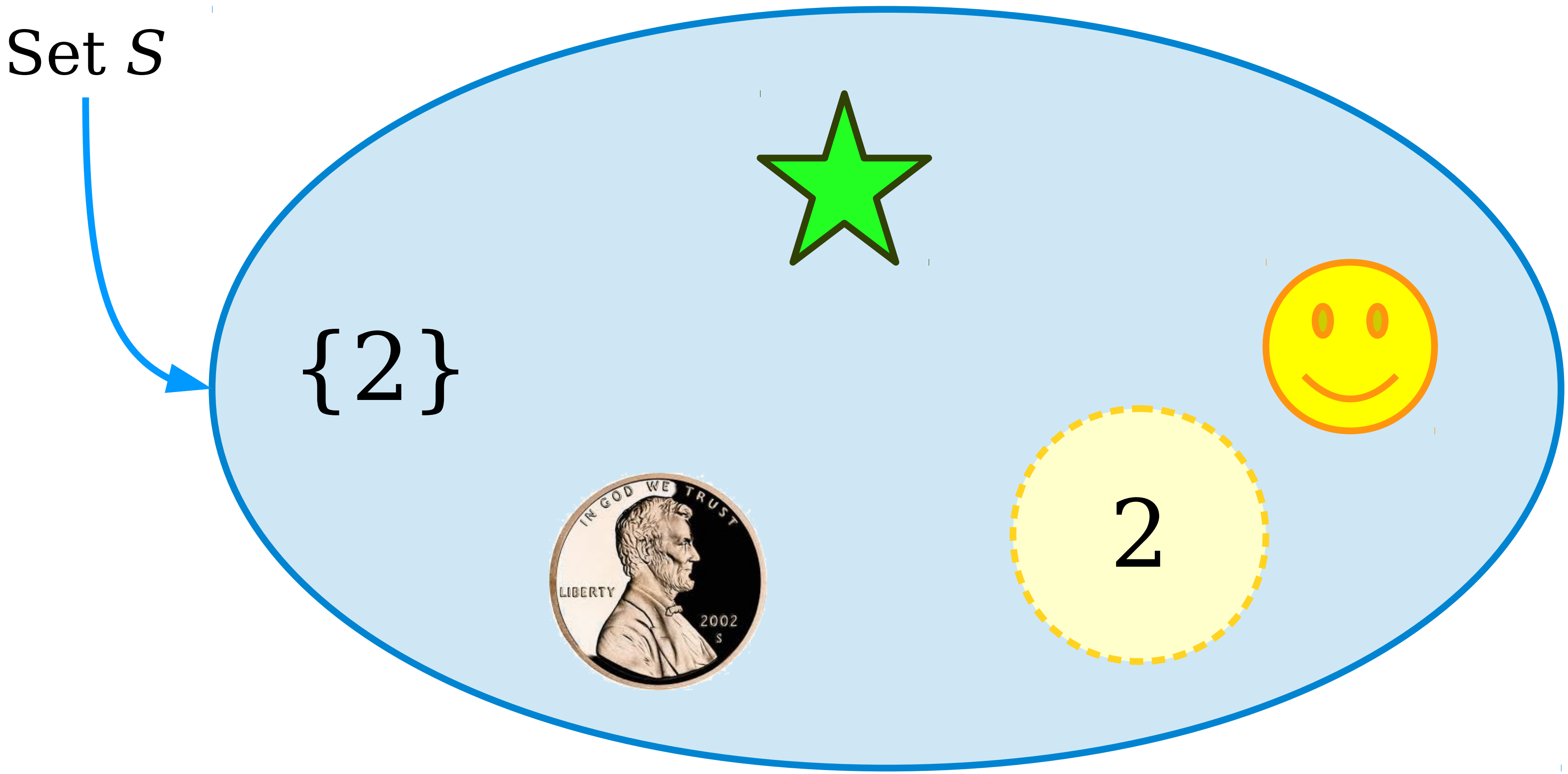


Subsets and Elements



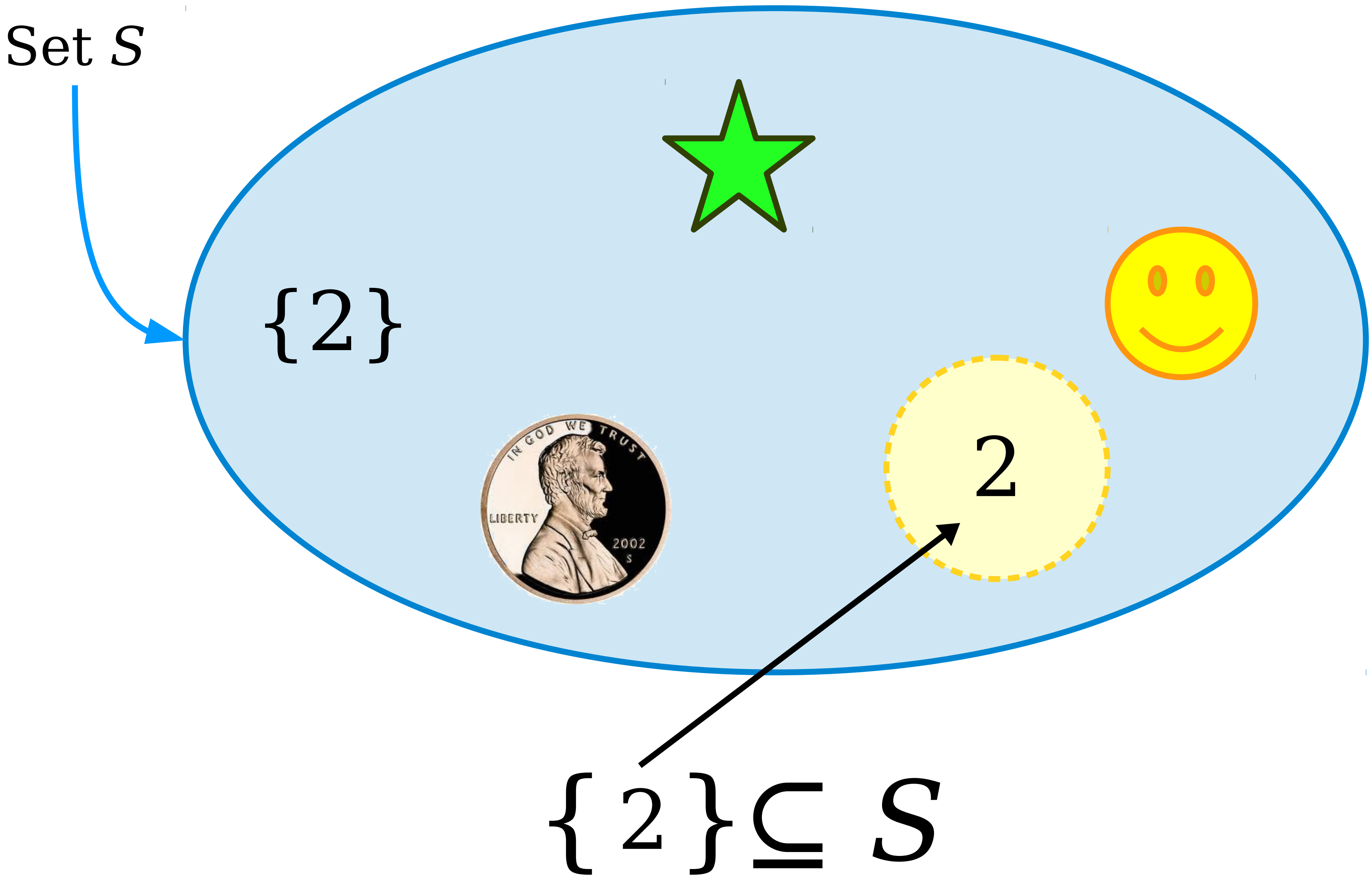
$$\{2\} \subseteq S$$

Subsets and Elements

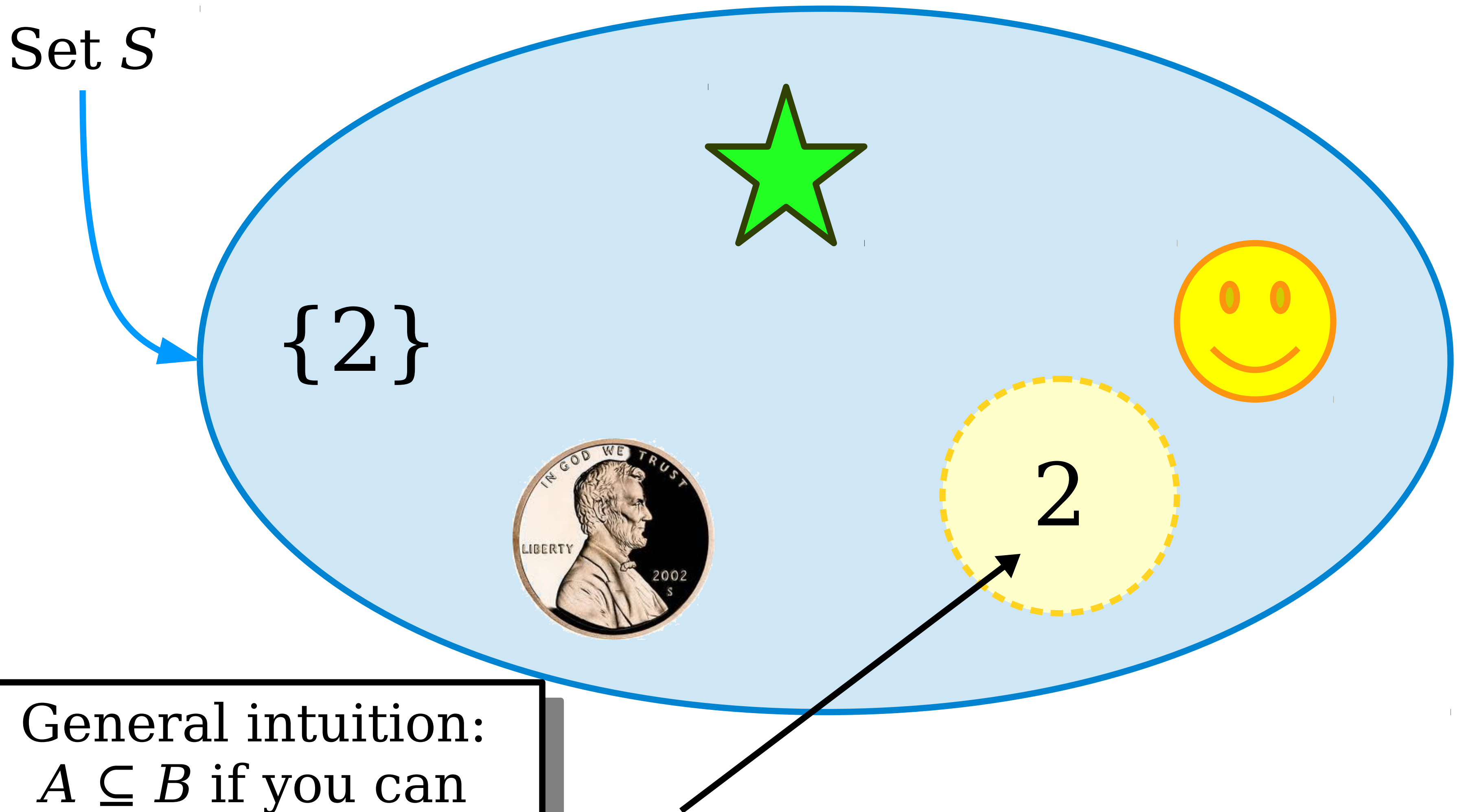


$$\{2\} \subseteq S$$

Subsets and Elements



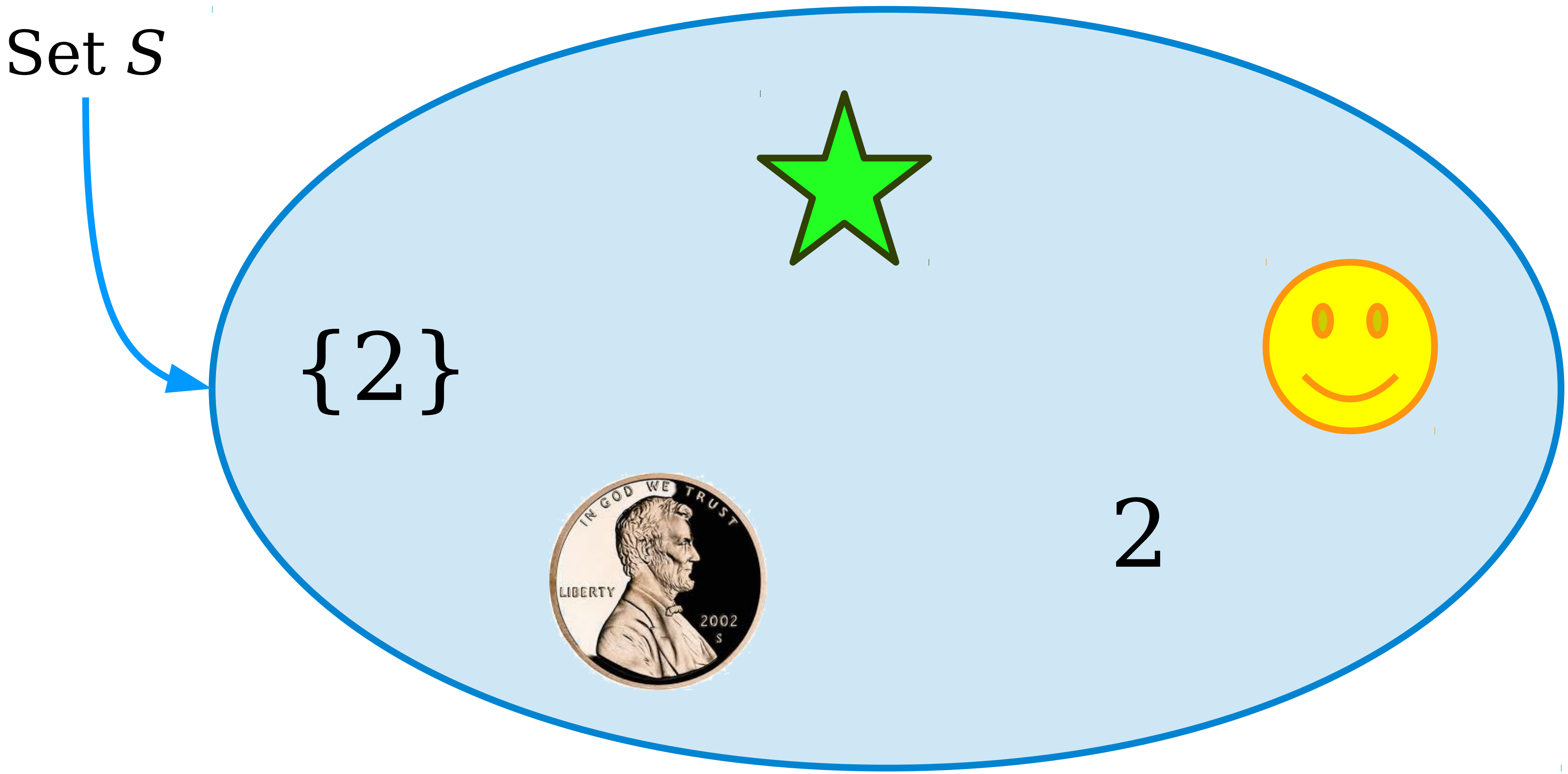
Subsets and Elements



General intuition:
$A \subseteq B$ if you can form A by ***circling elements of B***.

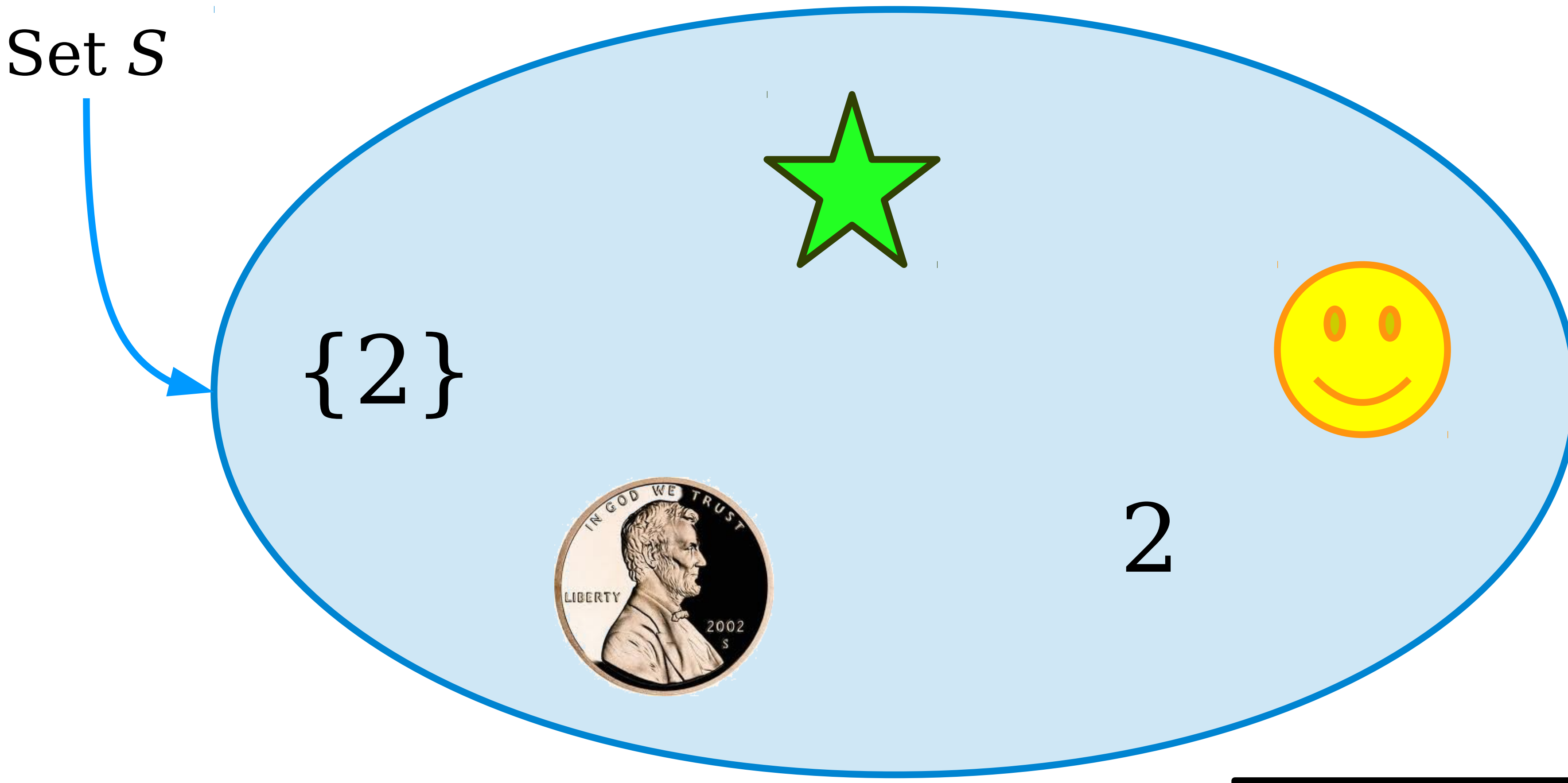
$$\{2\} \subseteq S$$

Subsets and Elements



$$2 \not\subseteq S$$

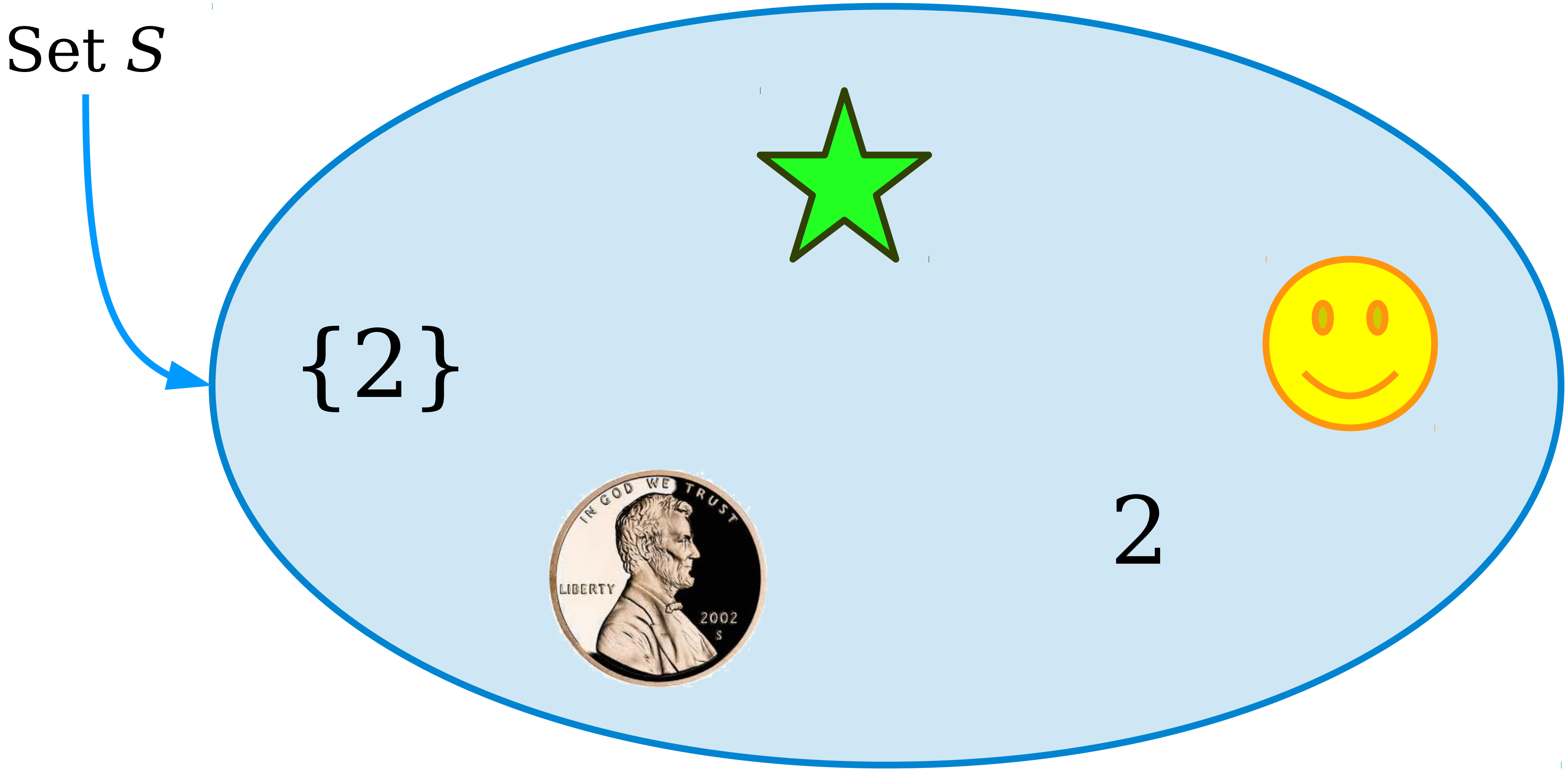
Subsets and Elements



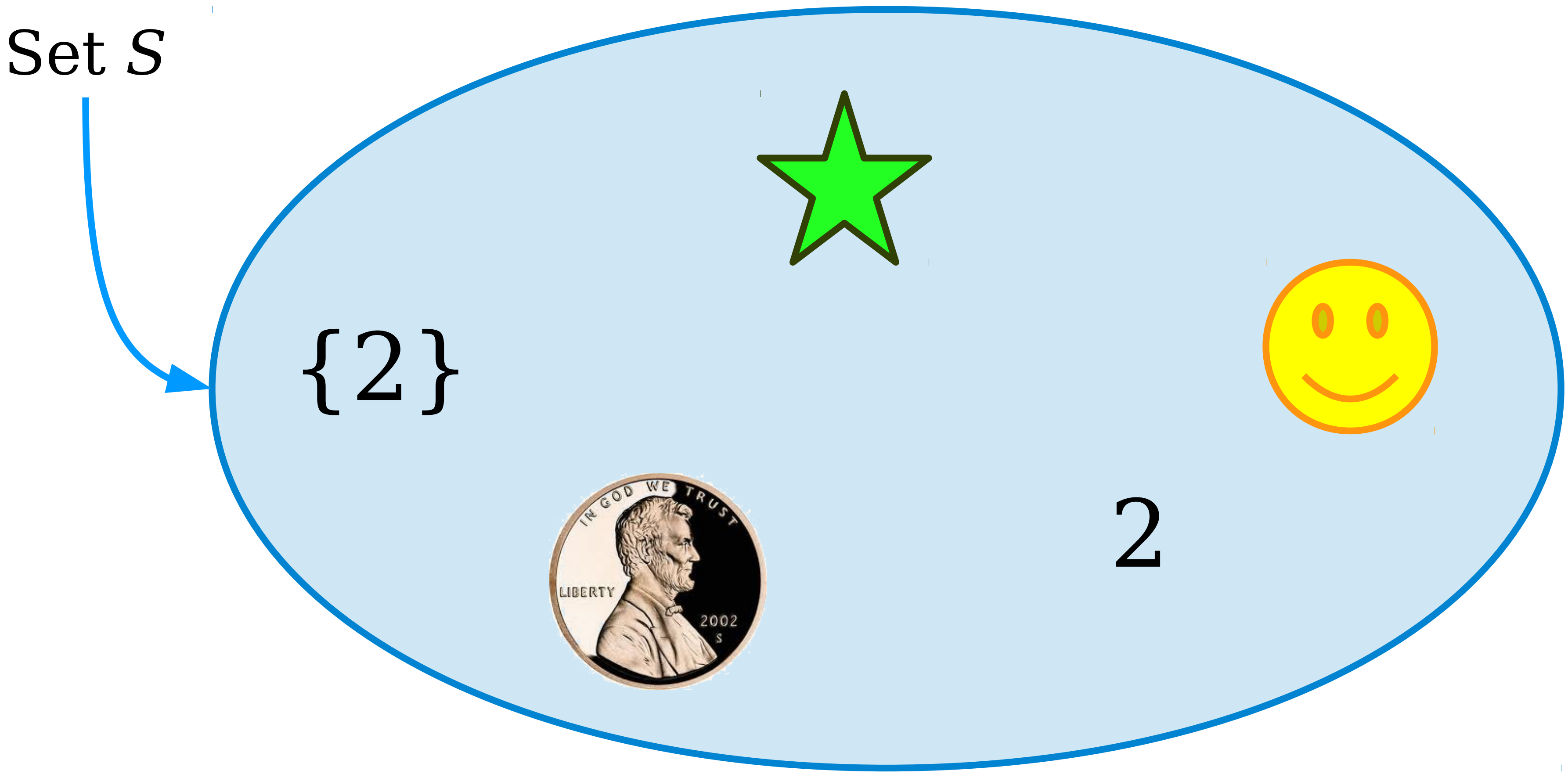
$$2 \not\subseteq S$$

(Since 2 isn't a set.)

Subsets and Elements

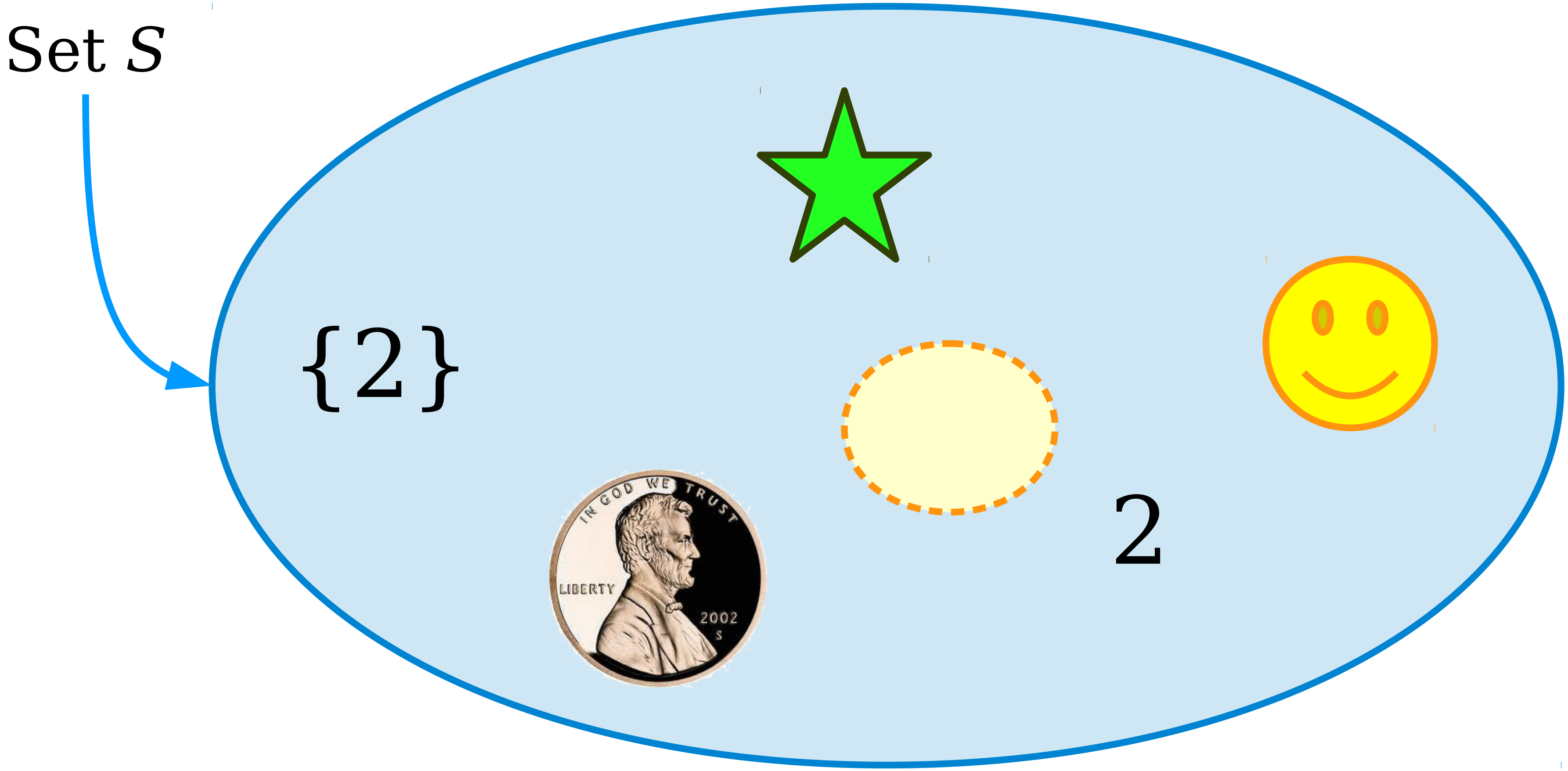


Subsets and Elements



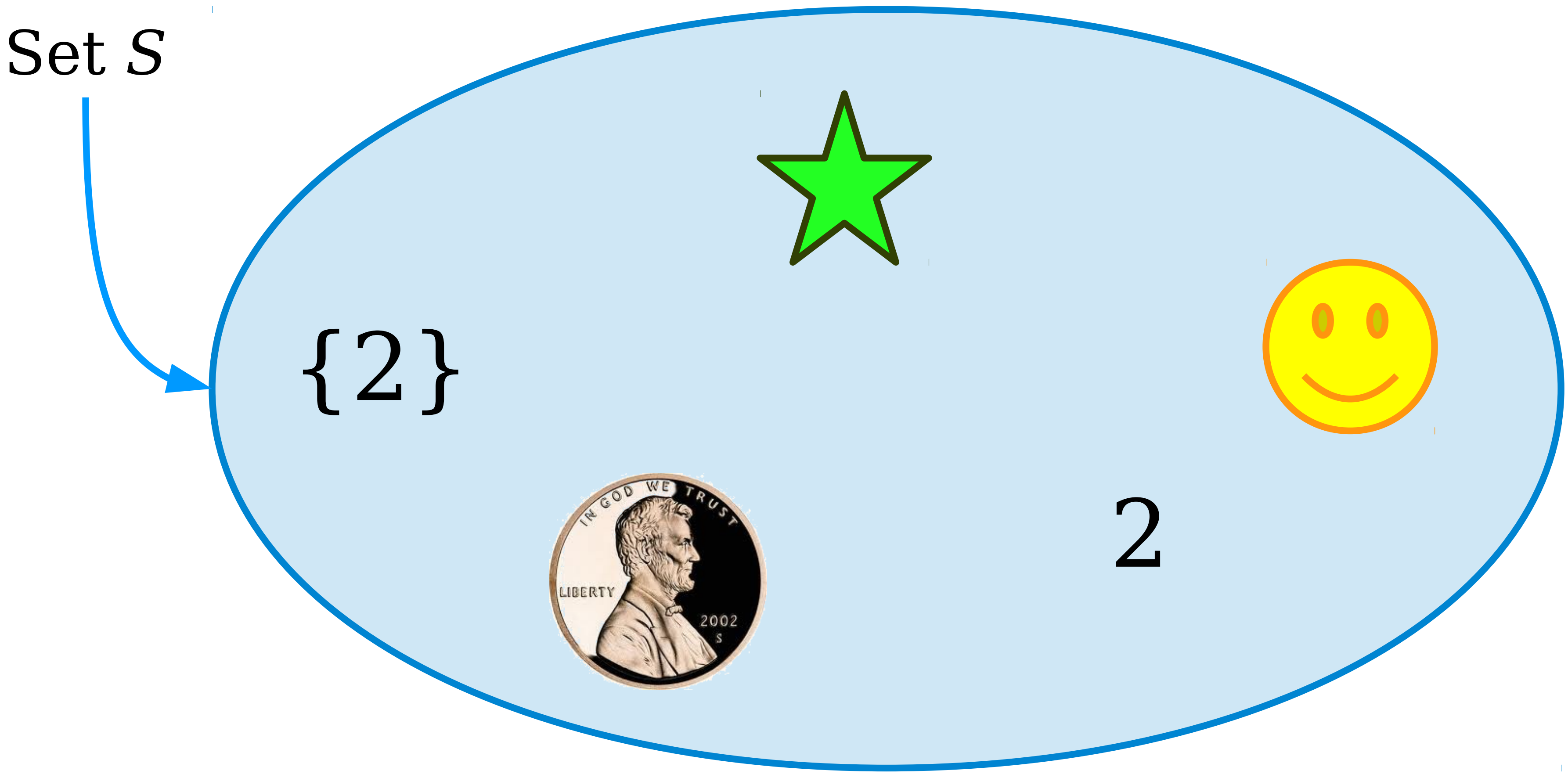
$$\emptyset \subseteq S$$

Subsets and Elements



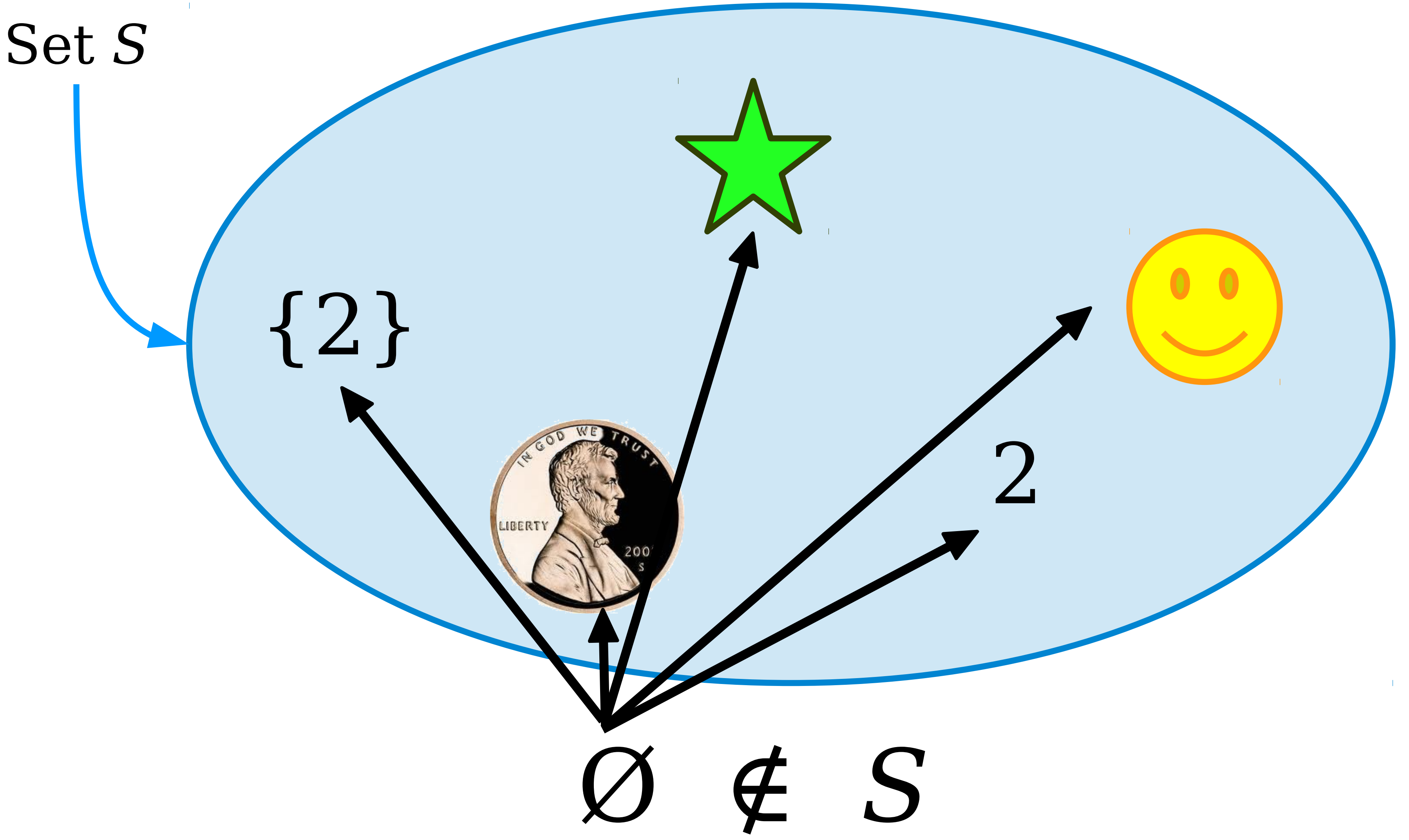
$$\varnothing \subseteq S$$

Subsets and Elements



$$\emptyset \notin S$$

Subsets and Elements



Subsets and Elements

- We say that $\boldsymbol{S \in T}$ if, among the elements of T, one of them is *exactly* the object S.
- We say that $\boldsymbol{S \subseteq T}$ 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.

$S = \{$ 

, 

$\}$

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

$\}$, $\{$ 

$\}$, $\{$ 

, 

$\}$ $\}$

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

Cardinality

Cardinality

- The ***cardinality*** of a set is the number of elements it contains.
- If S is a set, we denote its cardinality as $|S|$.
- Examples:
  - $|\{whimsy, mirth\}| = 2$
  - $|\{\{a, b\}, \{c, d, e, f, g\}, \{h\}\}| = 3$
  - $|\{1, 2, 3, 3, 3, 3, 3\}| = 3$
  - $|\{ n \in \mathbb{N} \mid n < 4 \}| = |\{0, 1, 2, 3\}| = 4$
  - $| \emptyset | = 0$
  - $| \{\emptyset\} | = 1$

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 $\mathbb{N}$

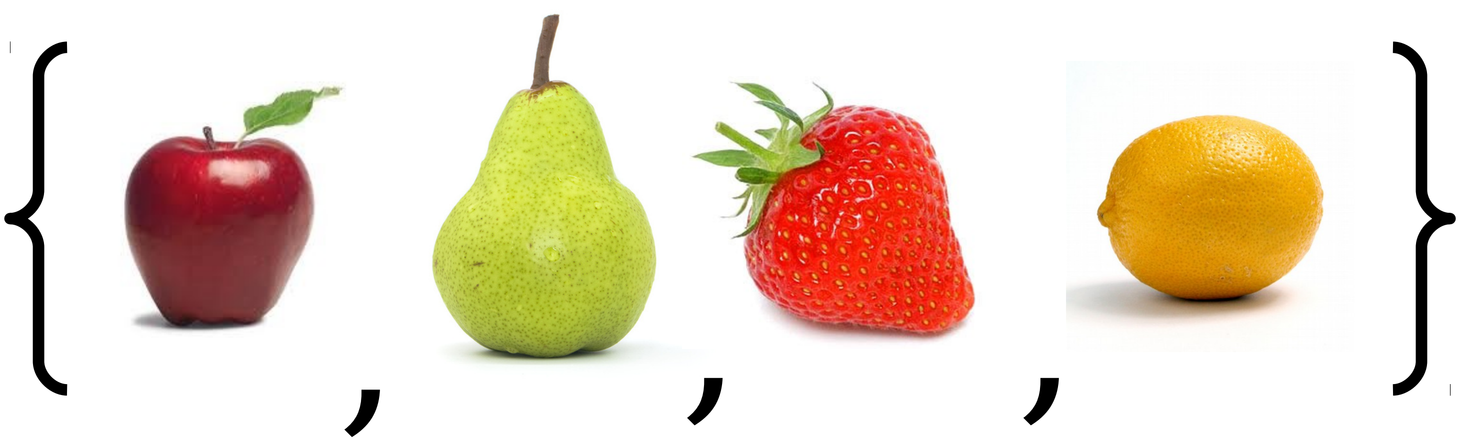
- What is $|\mathbb{N}|$?
  - There are infinitely many natural numbers.
  - $|\mathbb{N}|$ can't be a natural number, since it's infinitely large.
- We need to introduce a new term.
- Let's define $\boldsymbol{\aleph_0} = |\mathbb{N}|$.
  - $\aleph_0$ is pronounced “aleph-zero,” “aleph-nought,” or “aleph-null.”

Consider the set

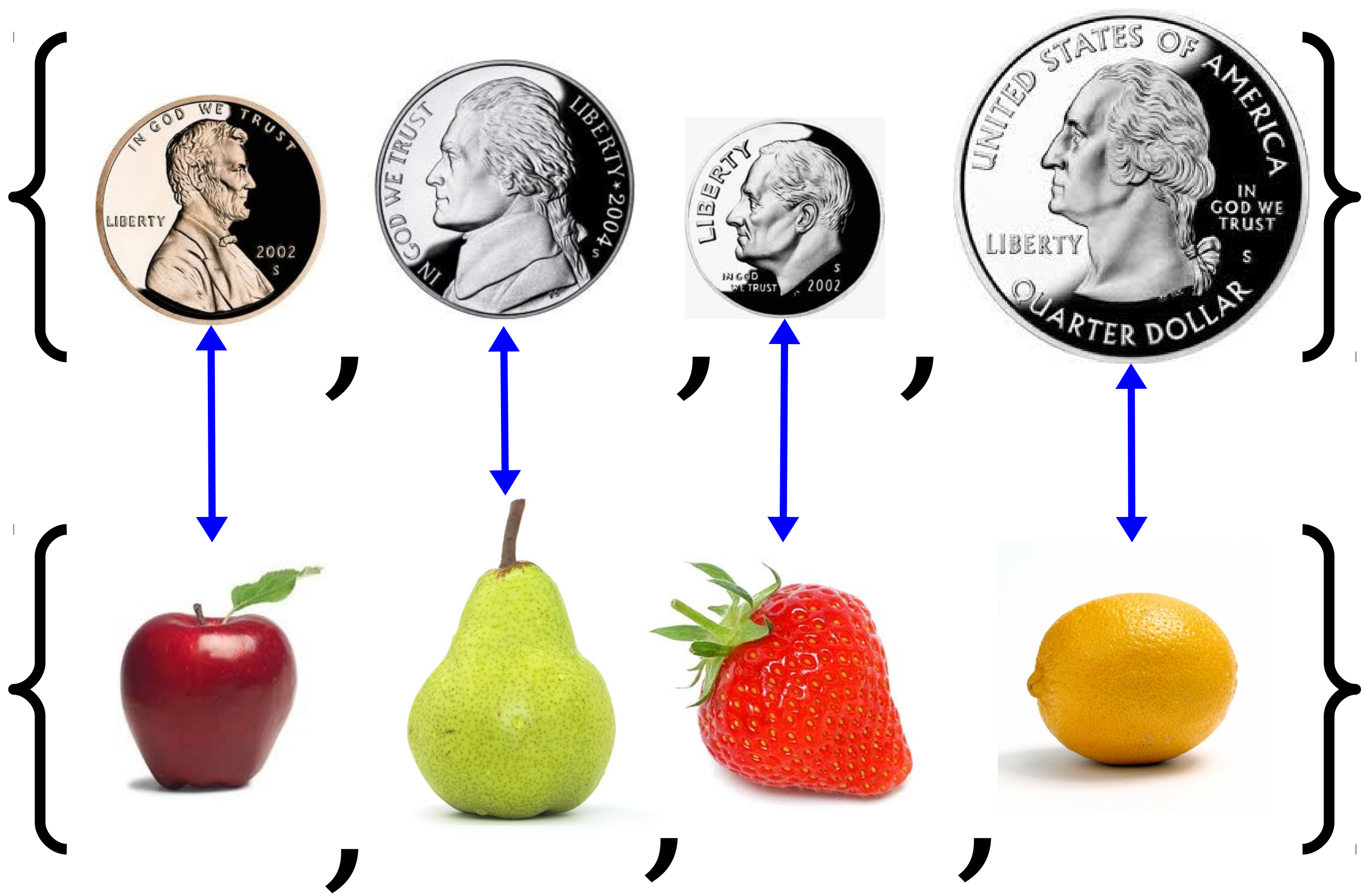
$$S = \{\ n \mid n \in \mathbb{N} \text{ and } n \text{ is even }\}.$$

What is $|S|$?

How Big Are These Sets?

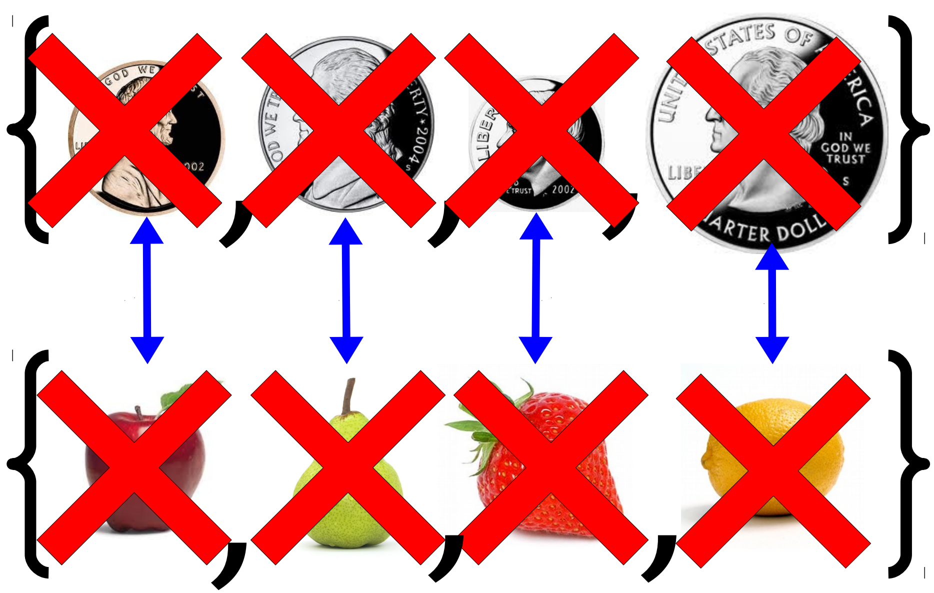


How Big Are These Sets?



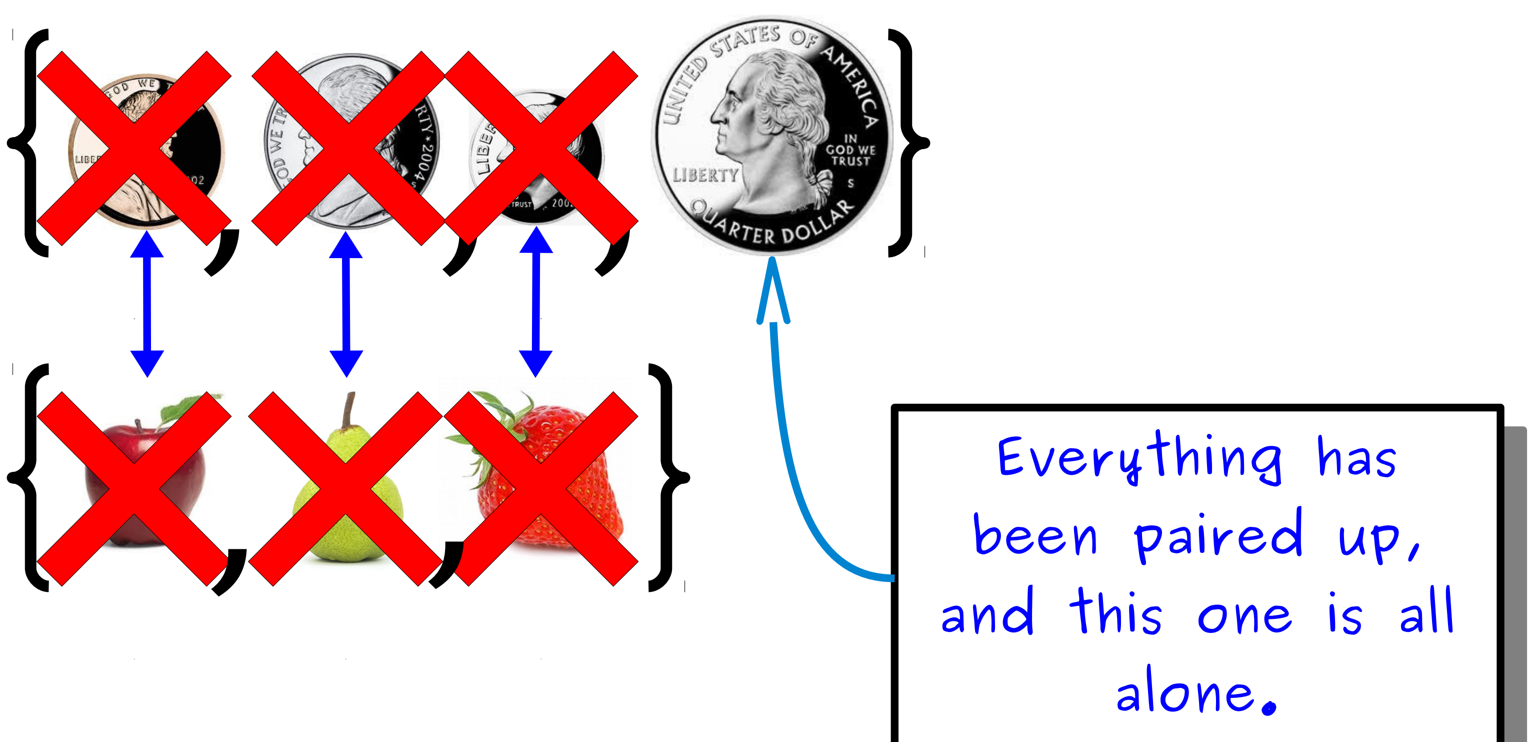
Comparing Cardinalities

- Two sets have the same cardinality if there is a way to pair their elements off without leaving any elements uncovered.
- The intuition:

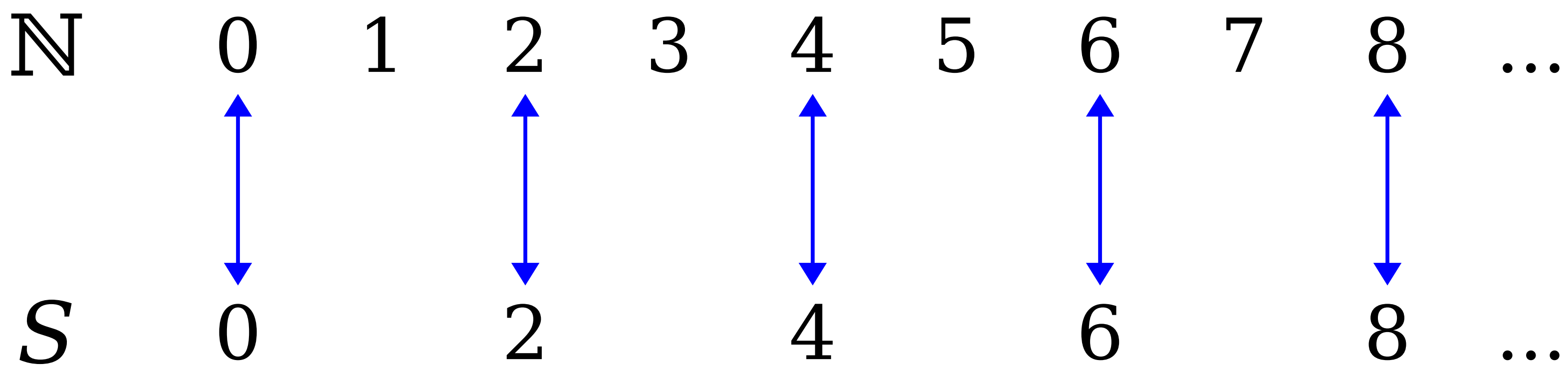


Comparing Cardinalities

- Two sets have the same cardinality if there is a way to pair their elements off without leaving any elements uncovered.
- The intuition:



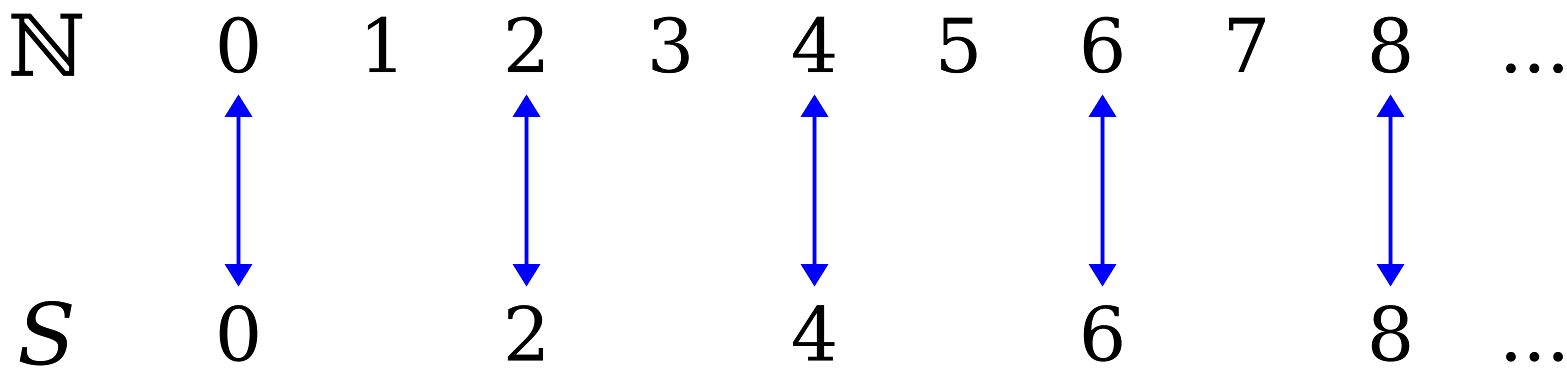
Infinite Cardinalities



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

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

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	...
	↕	↕	↕	↕	↕	↕	↕	↕	↕	
S	0	2	4	6	8	10	12	14	16	...

$$n \leftrightarrow 2n$$

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

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

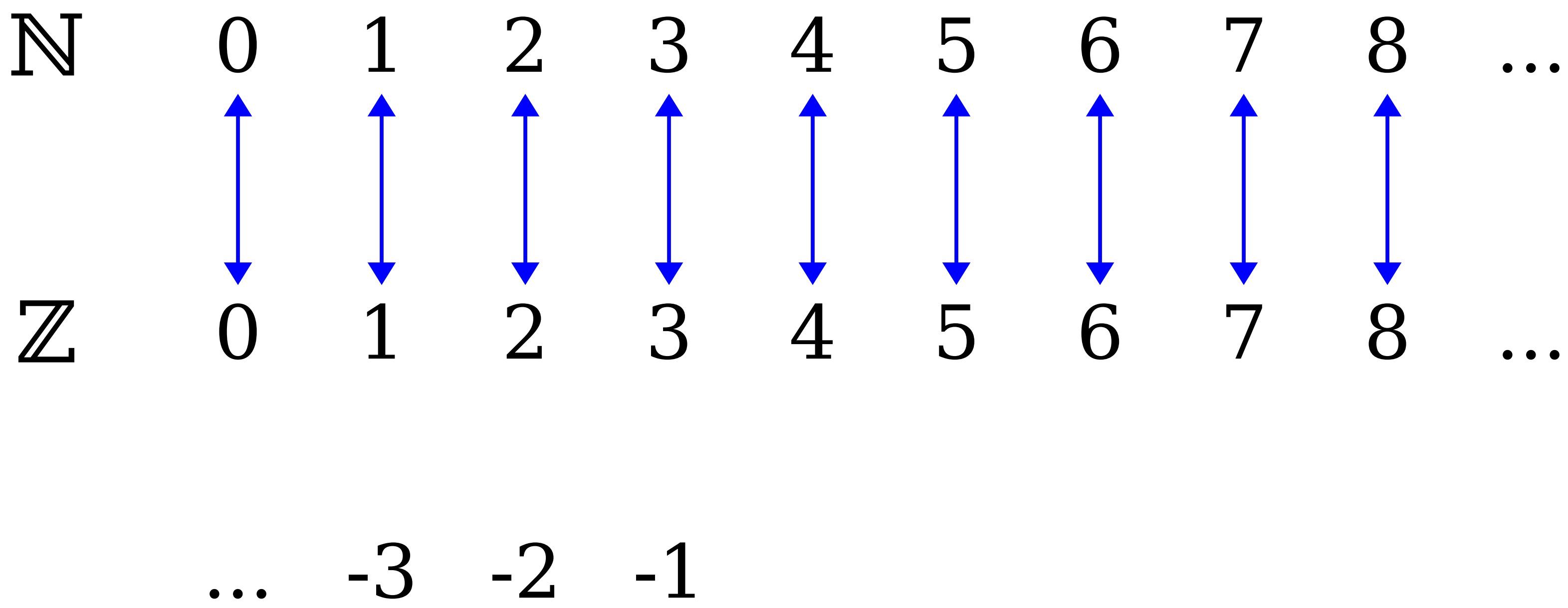
Infinite Cardinalities

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

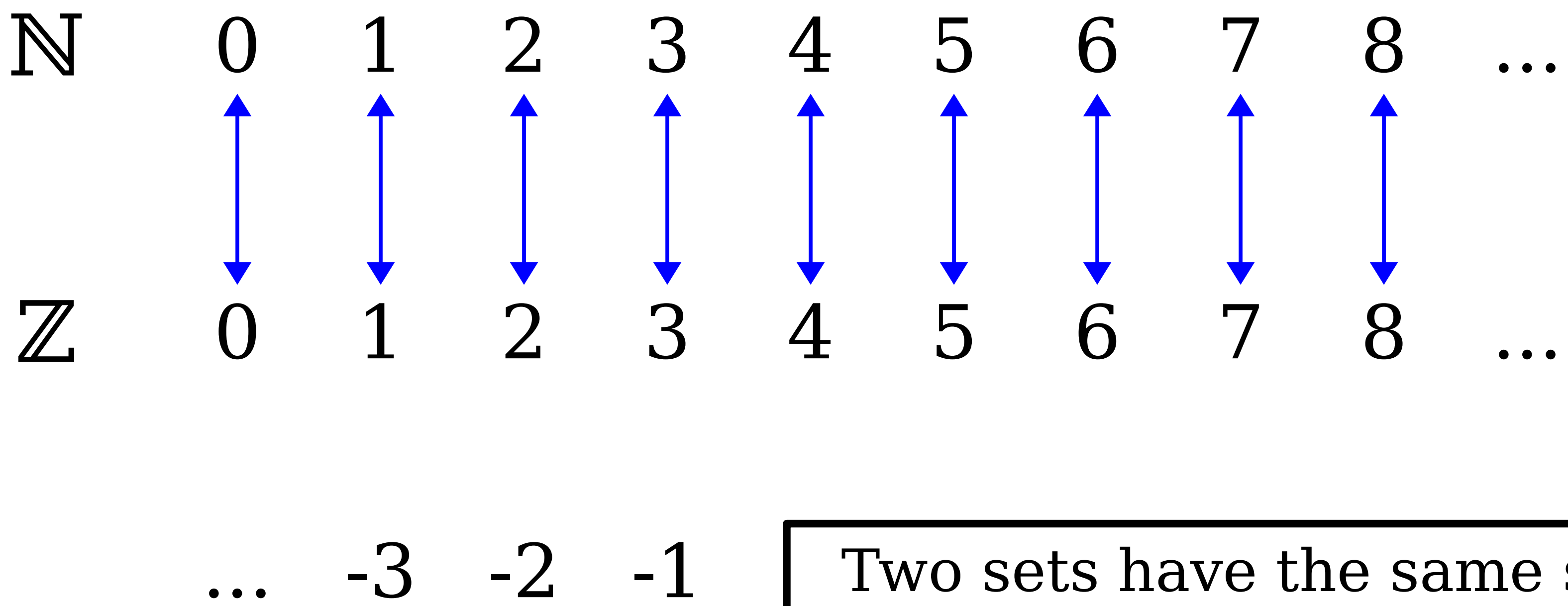
Infinite Cardinalities

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

Infinite Cardinalities



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	…

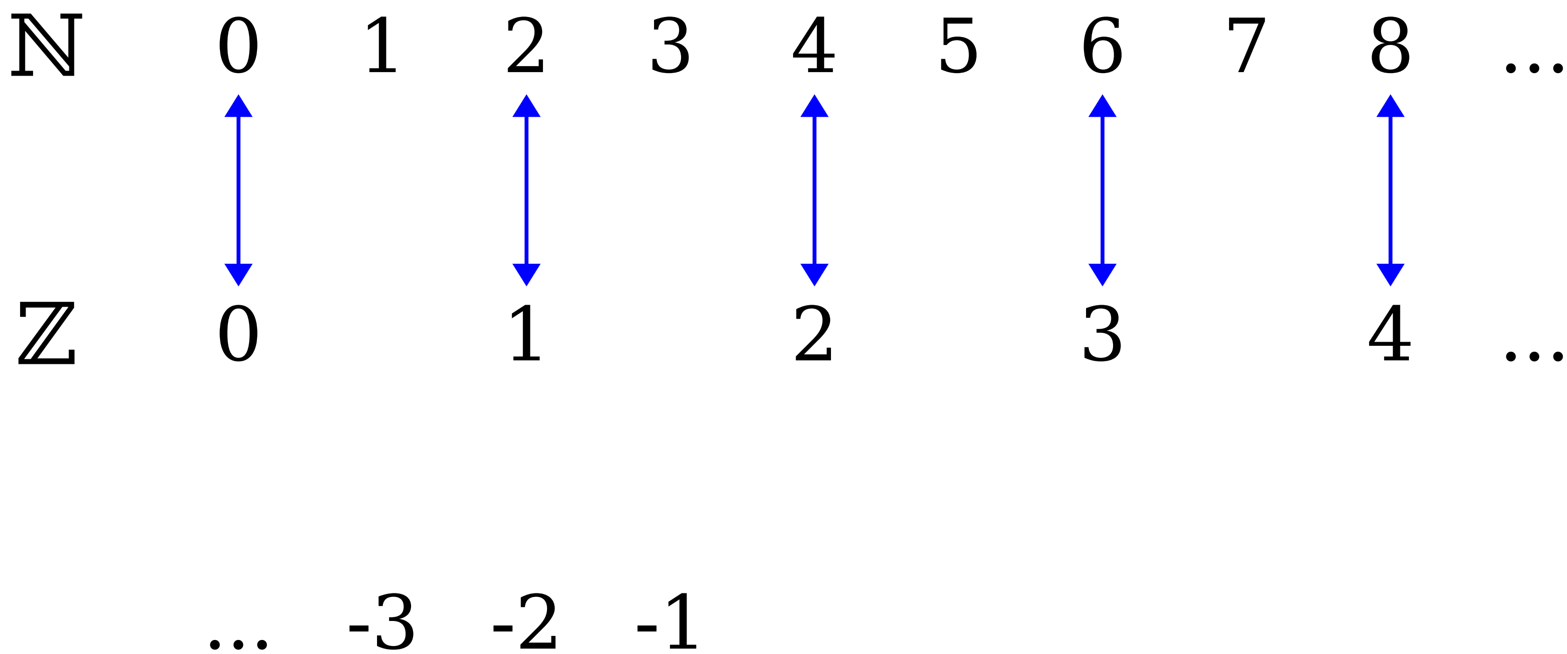
Infinite Cardinalities

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

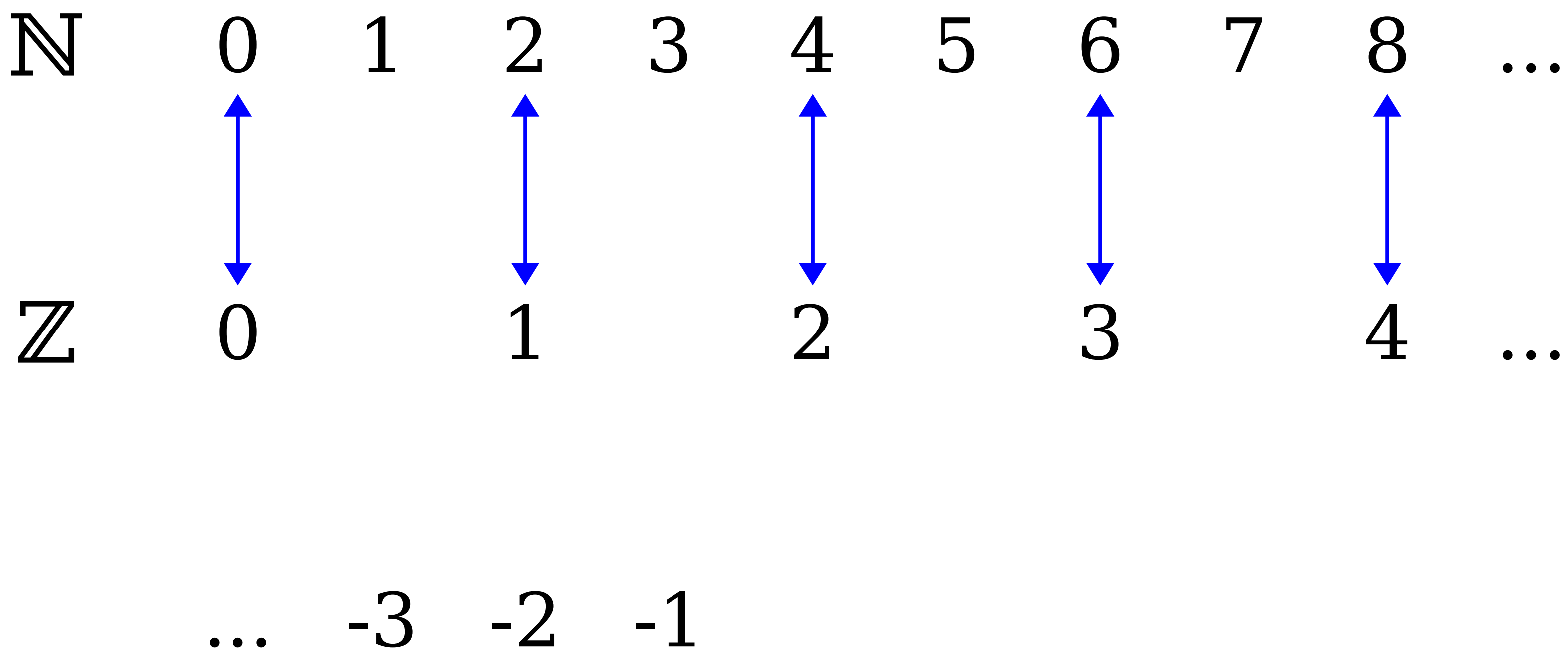
Infinite Cardinalities

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

Infinite Cardinalities

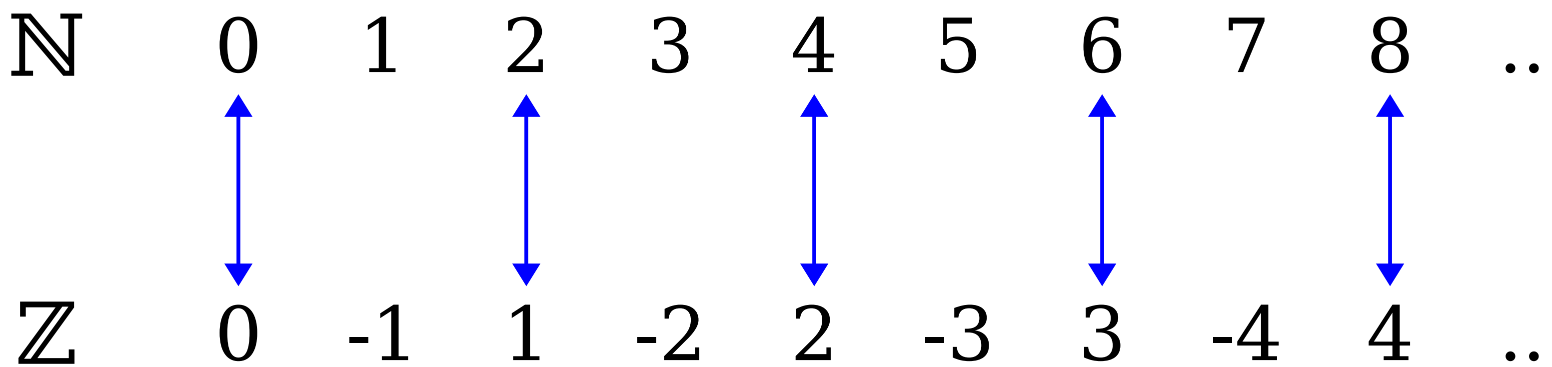


Infinite Cardinalities



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

$\mathbb{N}$	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.
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.

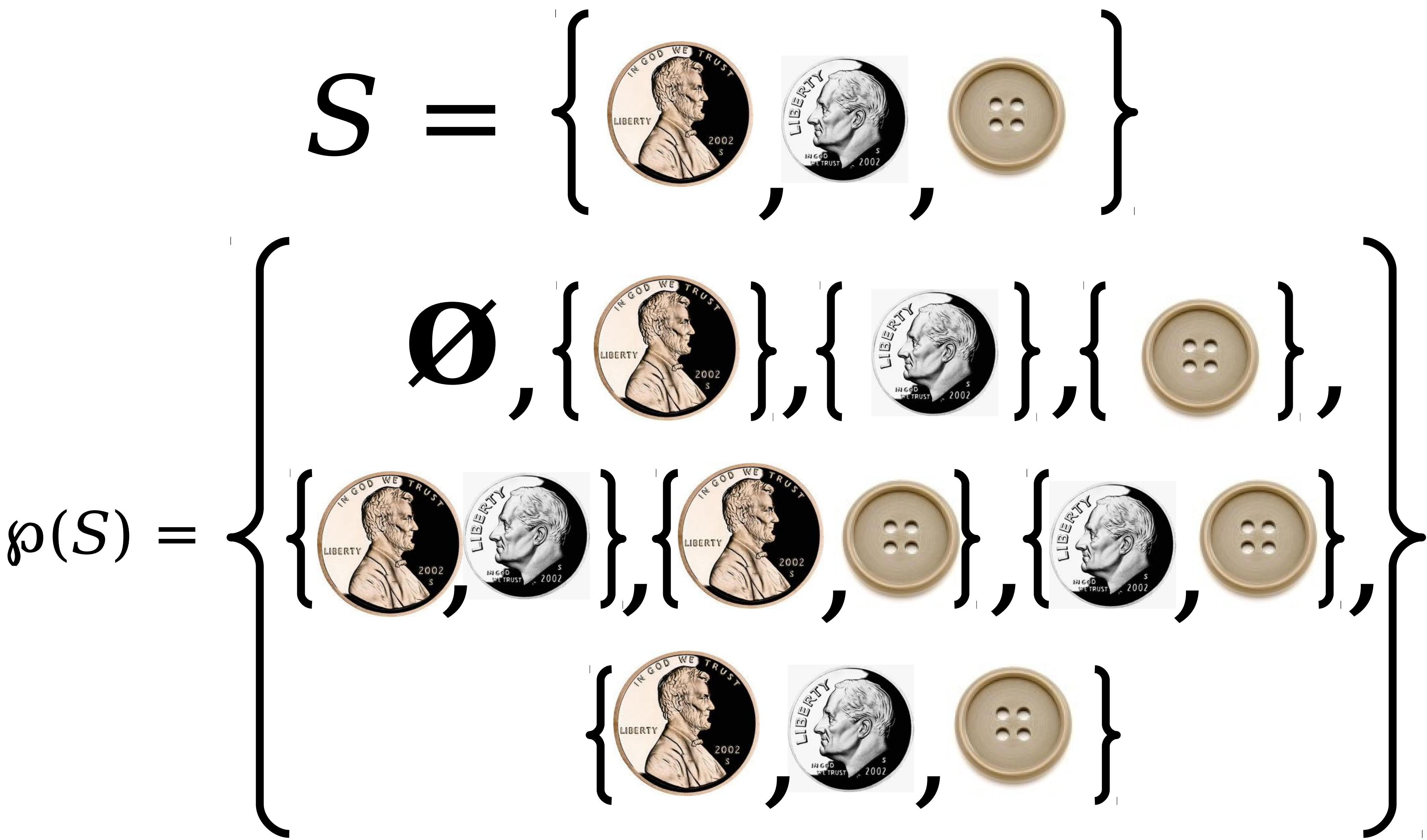
Important Question:

Do all infinite sets have the same cardinality?

$S = \{$  ,  $\}$

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

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



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

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

$$\wp(S) = \{$$

$$\varnothing,$$

$$\{\boldsymbol{a}\}, \{\boldsymbol{b}\}, \{\boldsymbol{c}\}, \{\boldsymbol{d}\},$$

$$\{\boldsymbol{a}, \boldsymbol{b}\}, \{\boldsymbol{a}, \boldsymbol{c}\}, \{\boldsymbol{a}, \boldsymbol{d}\}, \{\boldsymbol{b}, \boldsymbol{c}\}, \{\boldsymbol{b}, \boldsymbol{d}\}, \{\boldsymbol{c}, \boldsymbol{d}\}$$

$$\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}, \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{d}\}, \{\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}\}, \{\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}\},$$

$$\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}\}$$

$$\}$$

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

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.

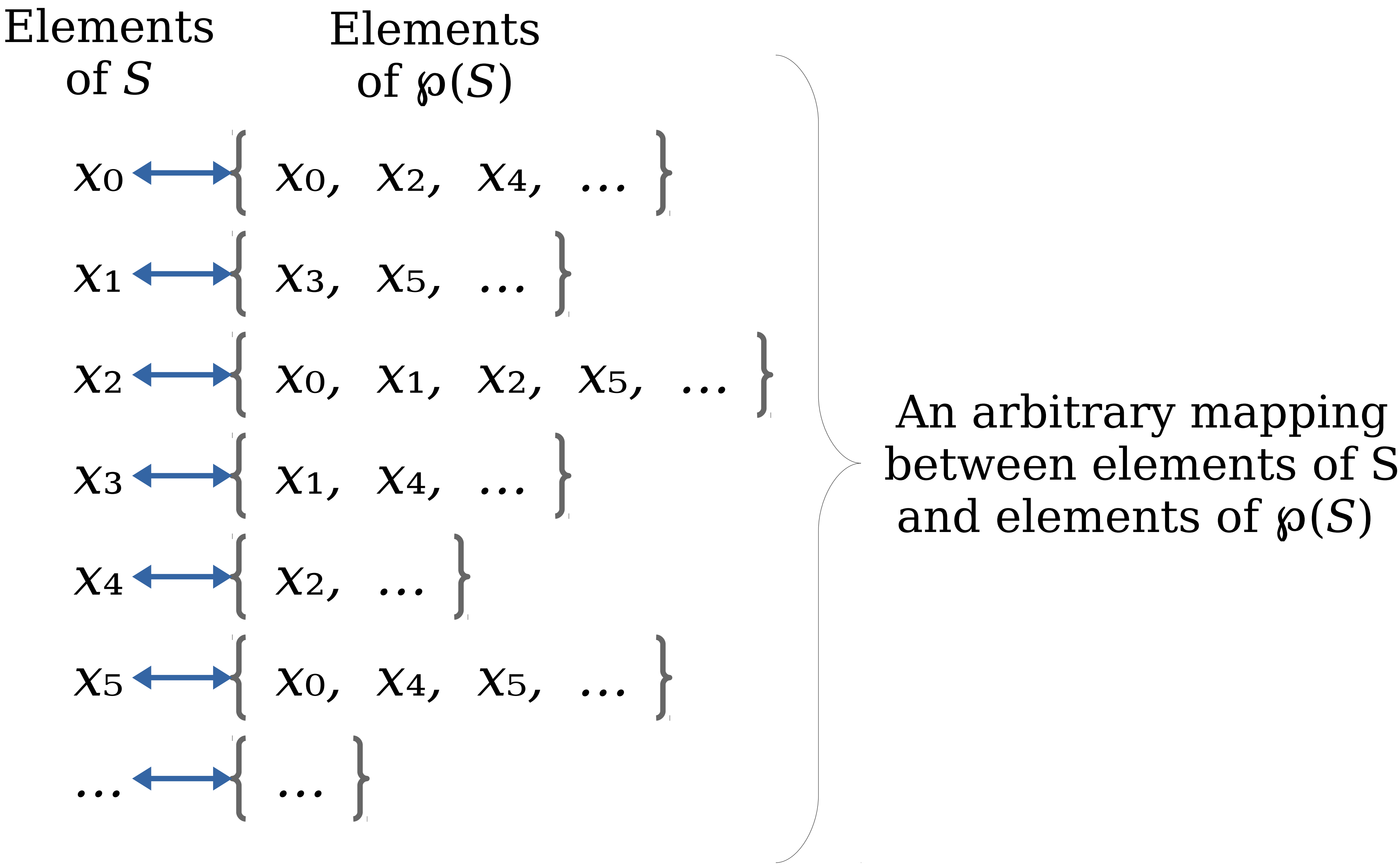
If $|S| = |\wp(S)|$, we can pair up the elements of S and the elements of $\wp(S)$ without leaving anything out.

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

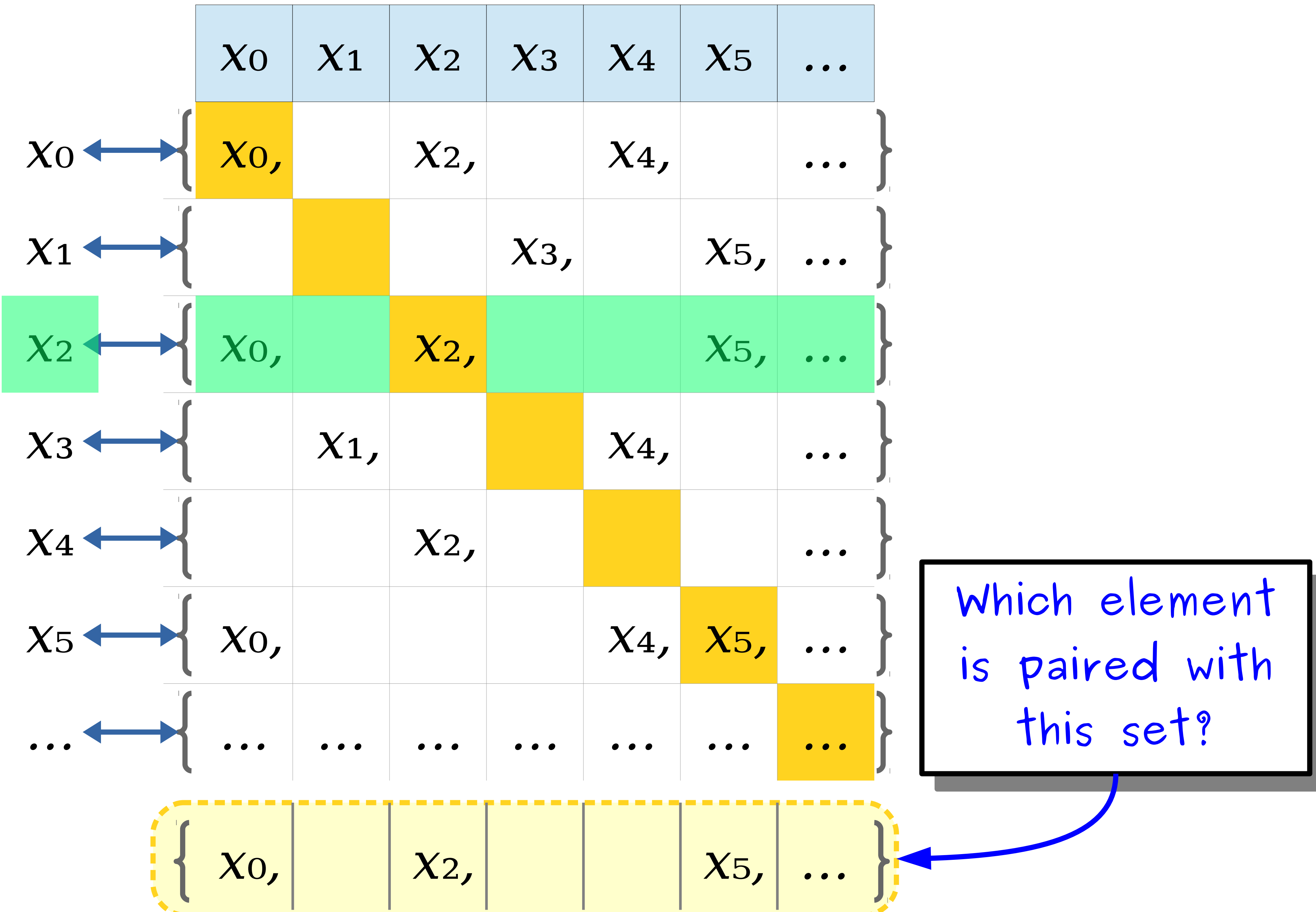
If $|S| = |\wp(S)|$, we can pair up the elements of S and the subsets of S without leaving anything out.

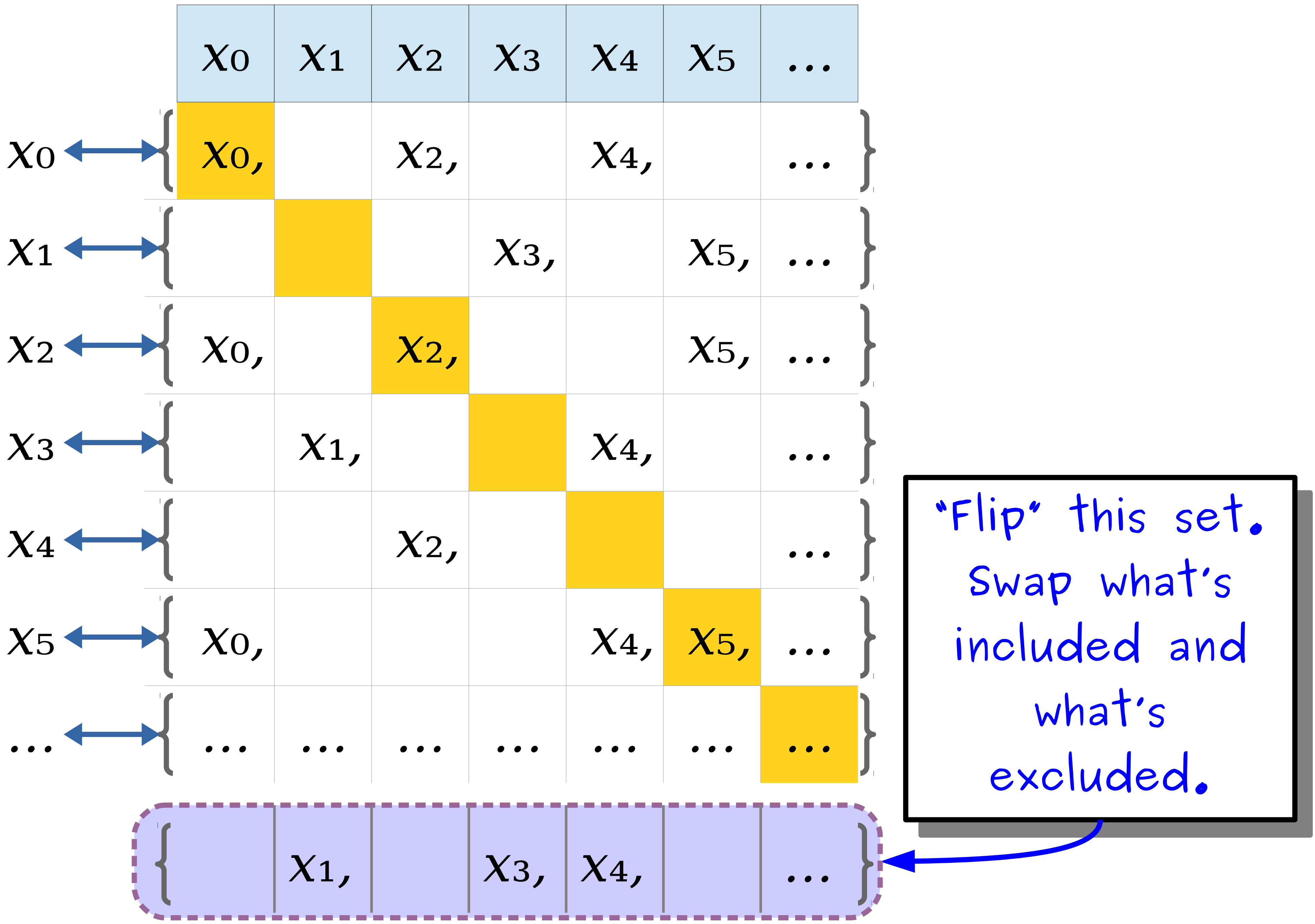
If $|S| = |\wp(S)|$, we can pair up the elements of S and the subsets of S without leaving anything out.

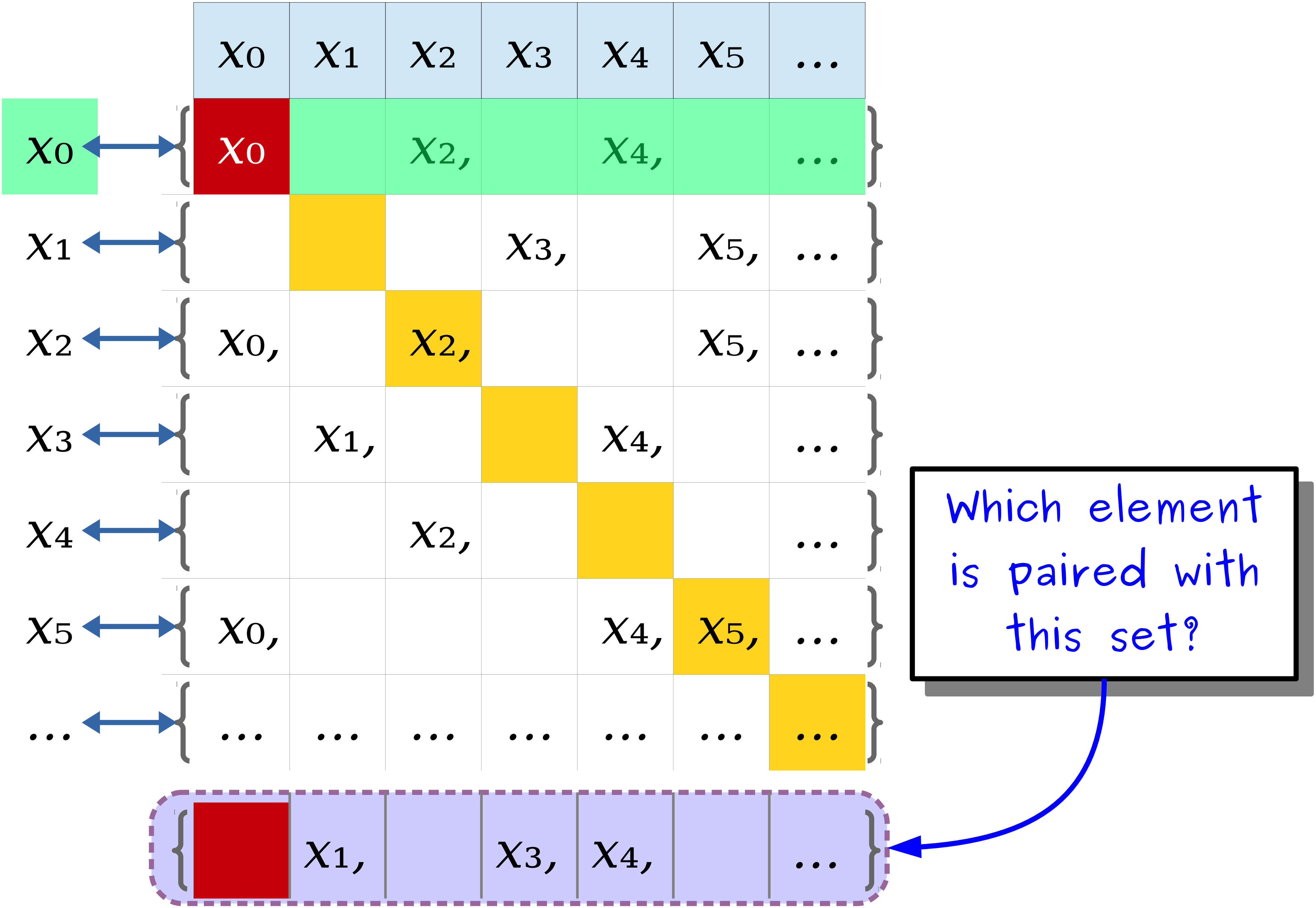
What would that look like?


Elements of S
Elements of $\wp(S)$
x_0 ⟷ $\{ x_0, x_2, x_4, \ldots \}$
x_1 ⟷ $\{ x_3, x_5, \ldots \}$
x_2 ⟷ $\{ x_0, x_1, x_2, x_5, \ldots \}$
x_3 ⟷ $\{ x_1, x_4, \ldots \}$
x_4 ⟷ $\{ x_2, \ldots \}$
x_5 ⟷ $\{ x_0, x_4, x_5, \ldots \}$
$\ldots$ ⟷ $\{ \ldots \}$
An arbitrary mapping between elements of S and elements of $\wp(S)$

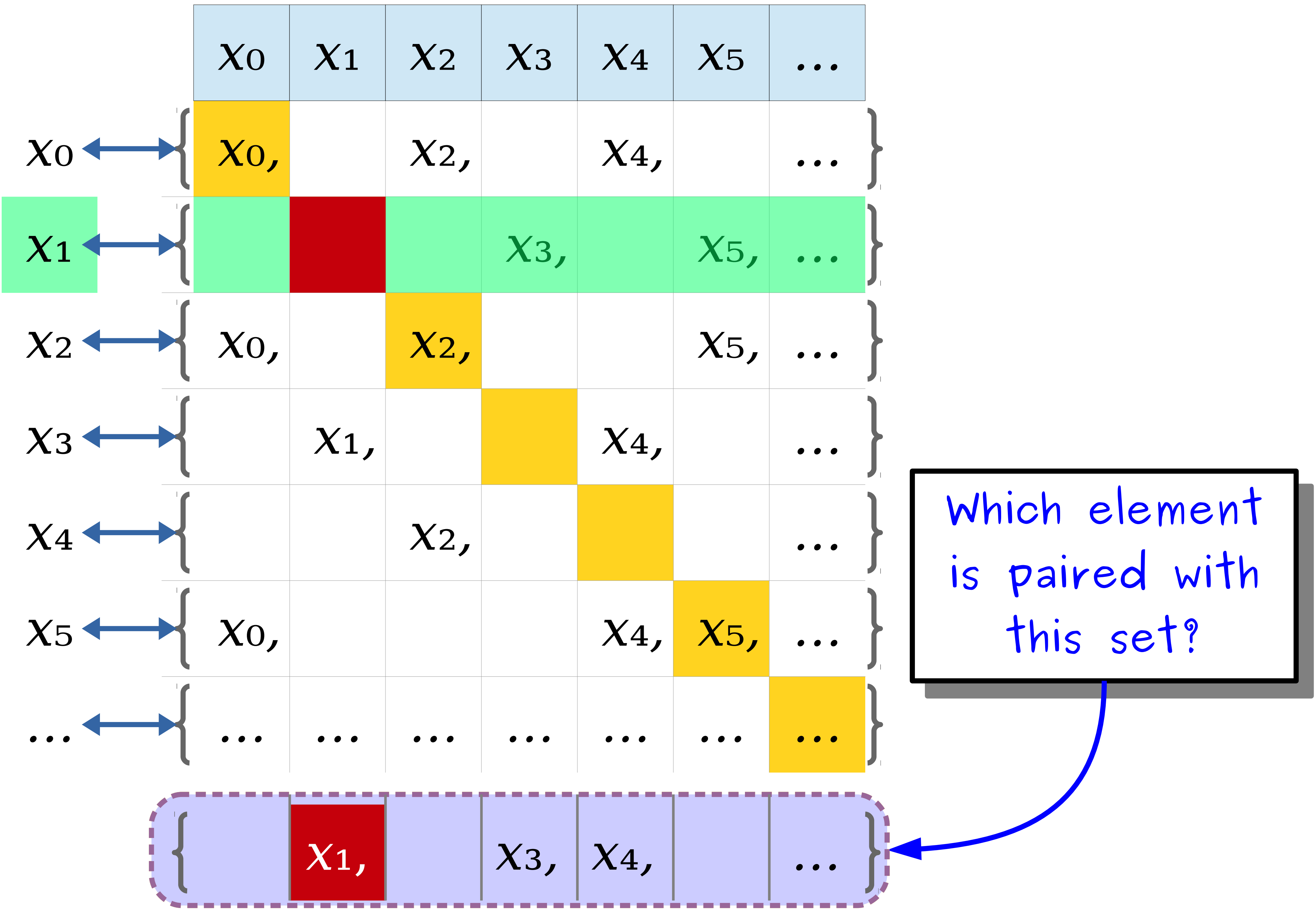
	x_0	x_1	x_2	x_3	x_4	x_5	...
$x_0 \leftrightarrow$	{ x_0,		x_2,		x_4,		... }
$x_1 \leftrightarrow$	{			x_3,		x_5,	... }
$x_2 \leftrightarrow$	{ x_0,		x_2,			x_5,	... }
$x_3 \leftrightarrow$	{	x_1,			x_4,		... }
$x_4 \leftrightarrow$	{		x_2,				... }
$x_5 \leftrightarrow$	{ x_0,				x_4,	x_5,	... }
... $\leftrightarrow$	{ ...	...	...	...	...	...	... }


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_0, x_2, x_5, ... }
Which element is paired with this set?

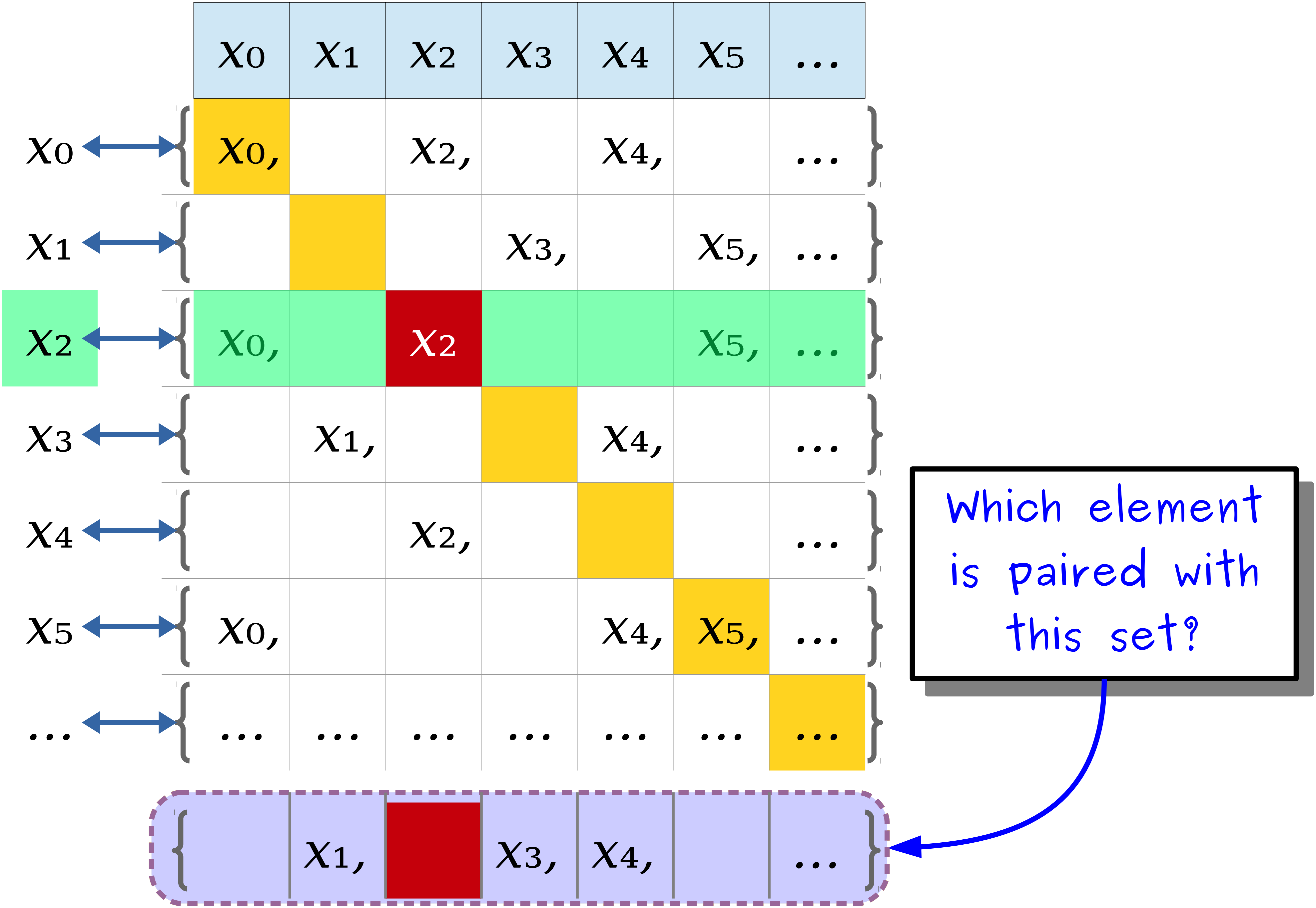

"Flip" this set. Swap what's included and what's excluded.

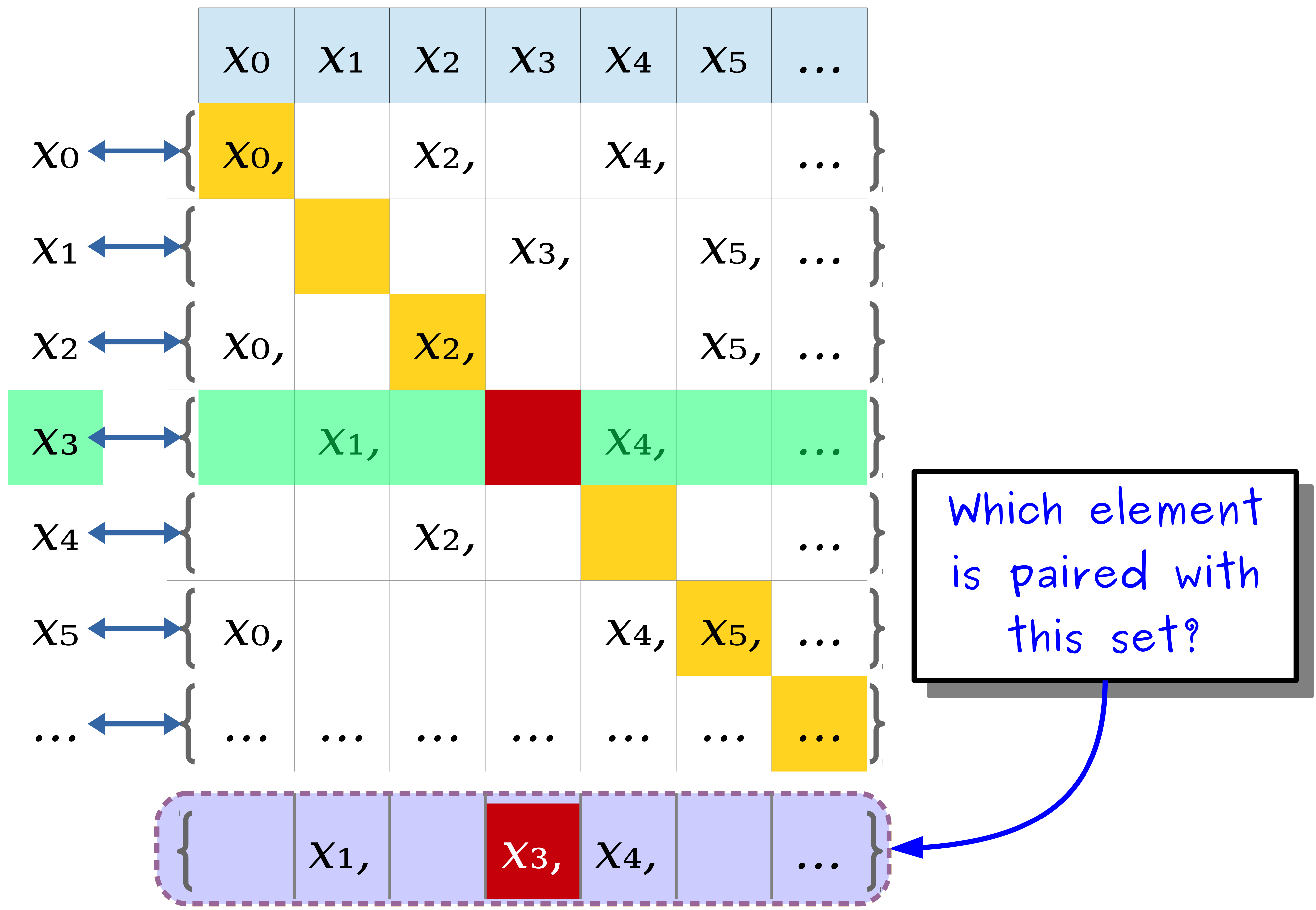


x_0 x_1 x_2 x_3 x_4 x_5 ...
x_0 ⟷ { x_0 x_2, x_4, ... }
x_1 ⟷ { x_3, x_5, ... }
x_2 ⟷ { x_0, x_2, x_5, ... }
x_3 ⟷ { x_1, x_4, ... }
x_4 ⟷ { x_2, ... }
x_5 ⟷ { x_0, x_4, x_5, ... }
... ⟷ { ... ... }
{ x_1, x_3, x_4, ... }
Which element is paired with this set?

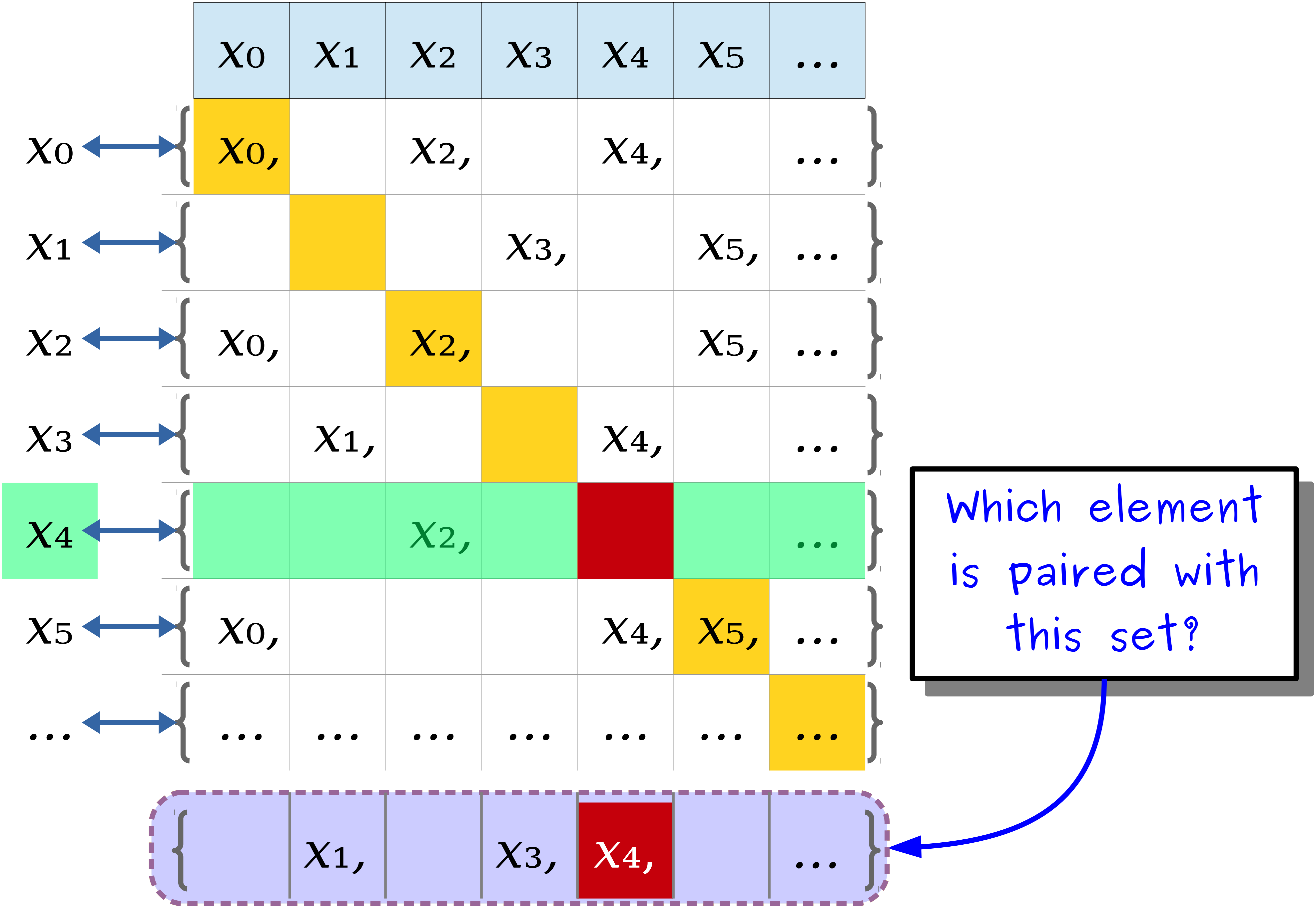


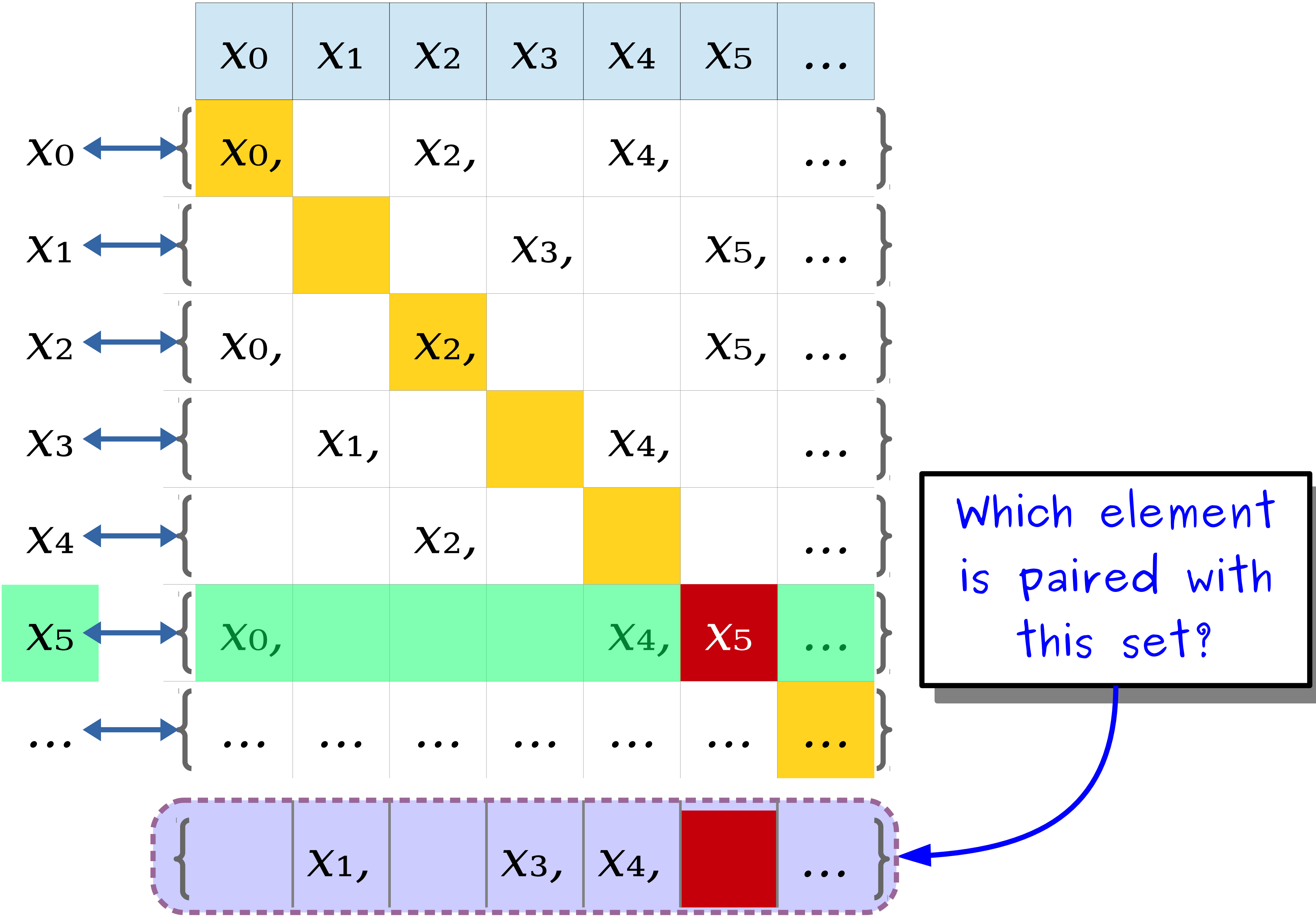
x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2, x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... ... }
{ x1, x3, x4, ... }
Which element is paired with this set?

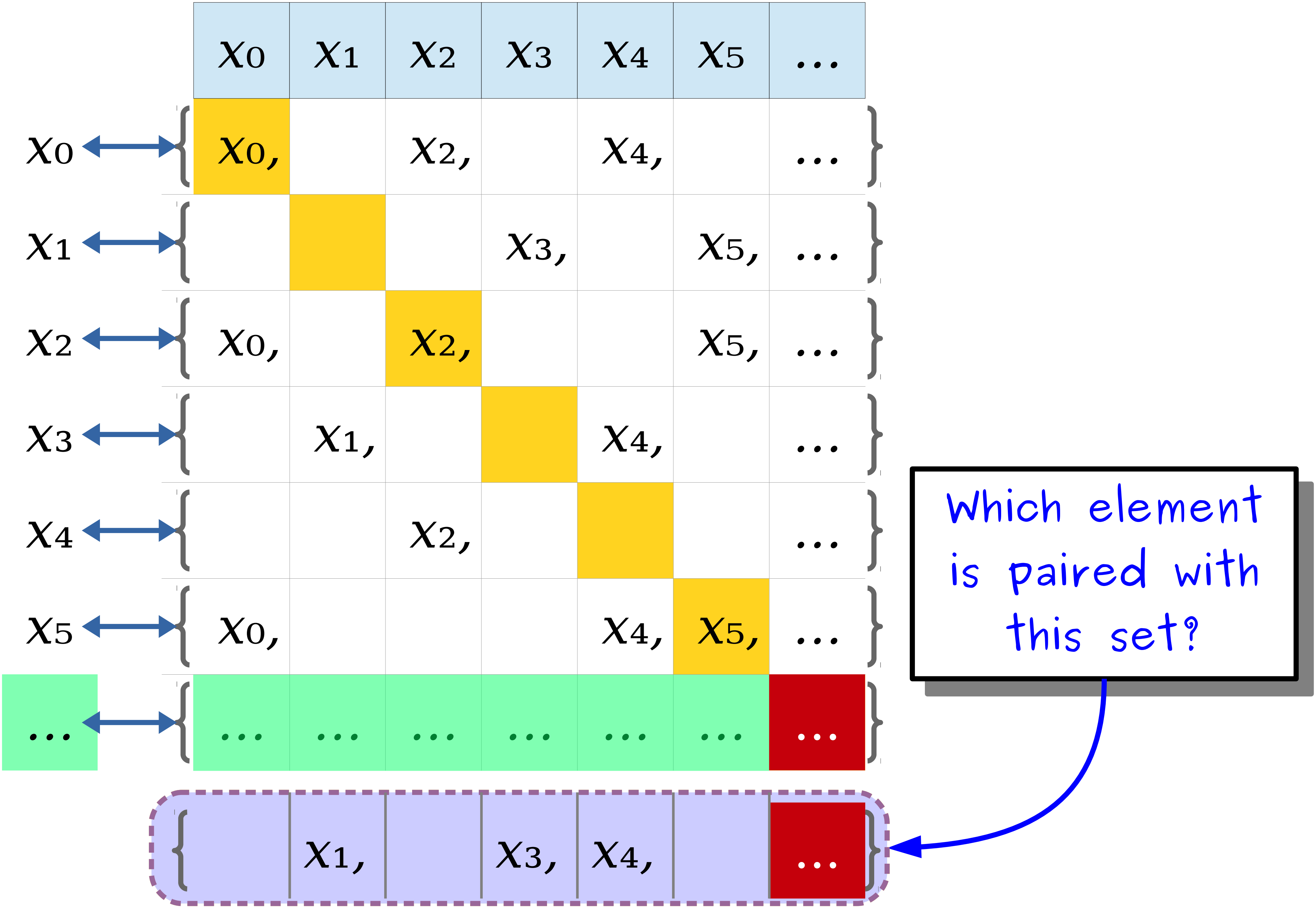

x0 x1 x2 x3 x4 x5 ...
x0 { x0, x2, x4, ... }
x1 { x3, x5, ... }
x2 { x0, x2 x5, ... }
x3 { x1, x4, ... }
x4 { x2, ... }
x5 { x0, x4, x5, ... }
... { ... ... }
{ x1, x3, x4, ... }
Which element is paired with this set?



Which element is paired with this set?


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?

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

Two Infinities…

- By Cantor's Theorem:

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

...And Beyond!

- By Cantor's Theorem:

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

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

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

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

$$\dots$$

- ***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.

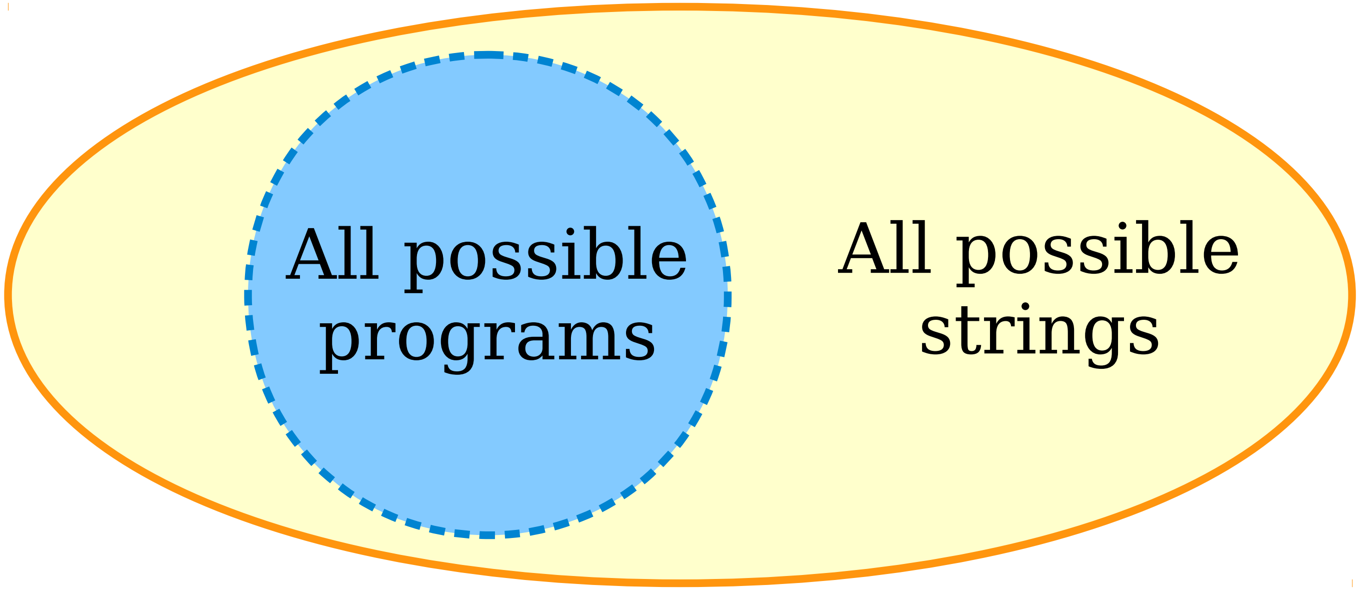
We're going to prove the following results:

- 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 = { "0", "1", "2", …, "9", "10", "11", … }

- 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

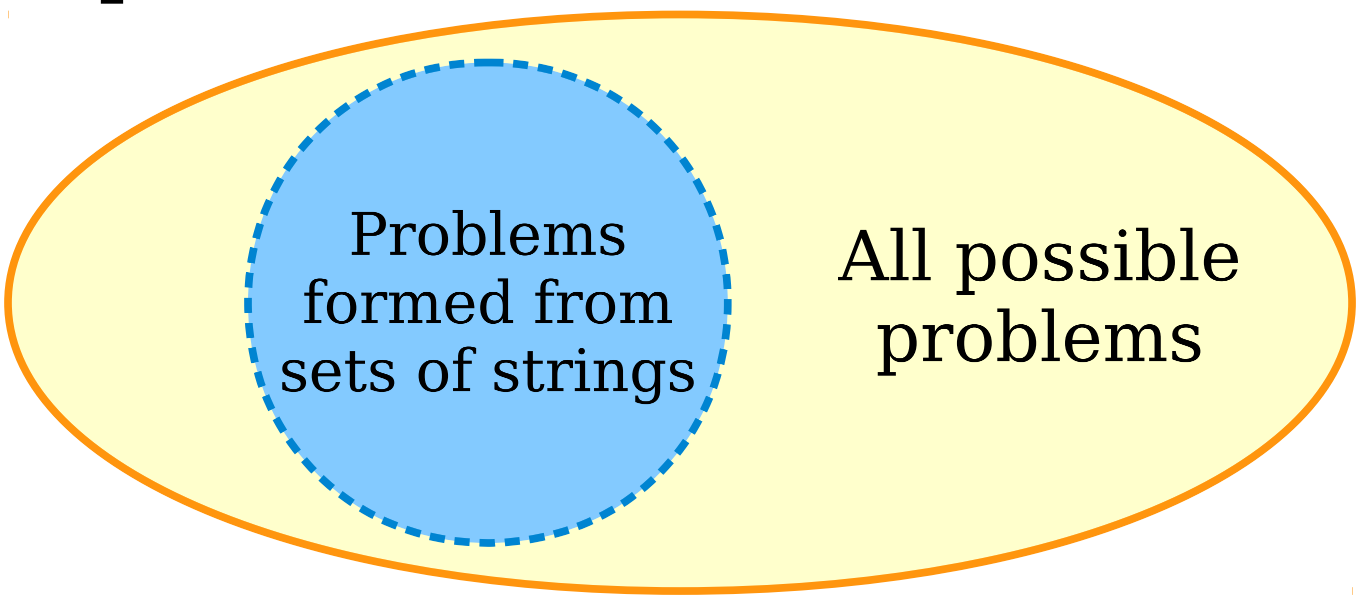
$$S = \{ p \mid p \text{ is a legal C++ program } \}$$

- 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!

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.

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

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?