CSE 311 Section 4

Set Theory & Structural Induction

Announcements & Reminders

- Homework 4 Part 1 due today (5/7) @ **6:00 pm** on Cozy
- Homework 4 Part 2 due Monday (5/11) @ **6:00 pm** on Gradescope
- Quiz 4 next week on Thursday (5/14)
  - Please review the homework feedback before quizzes

Sets: Quick Review

Sets

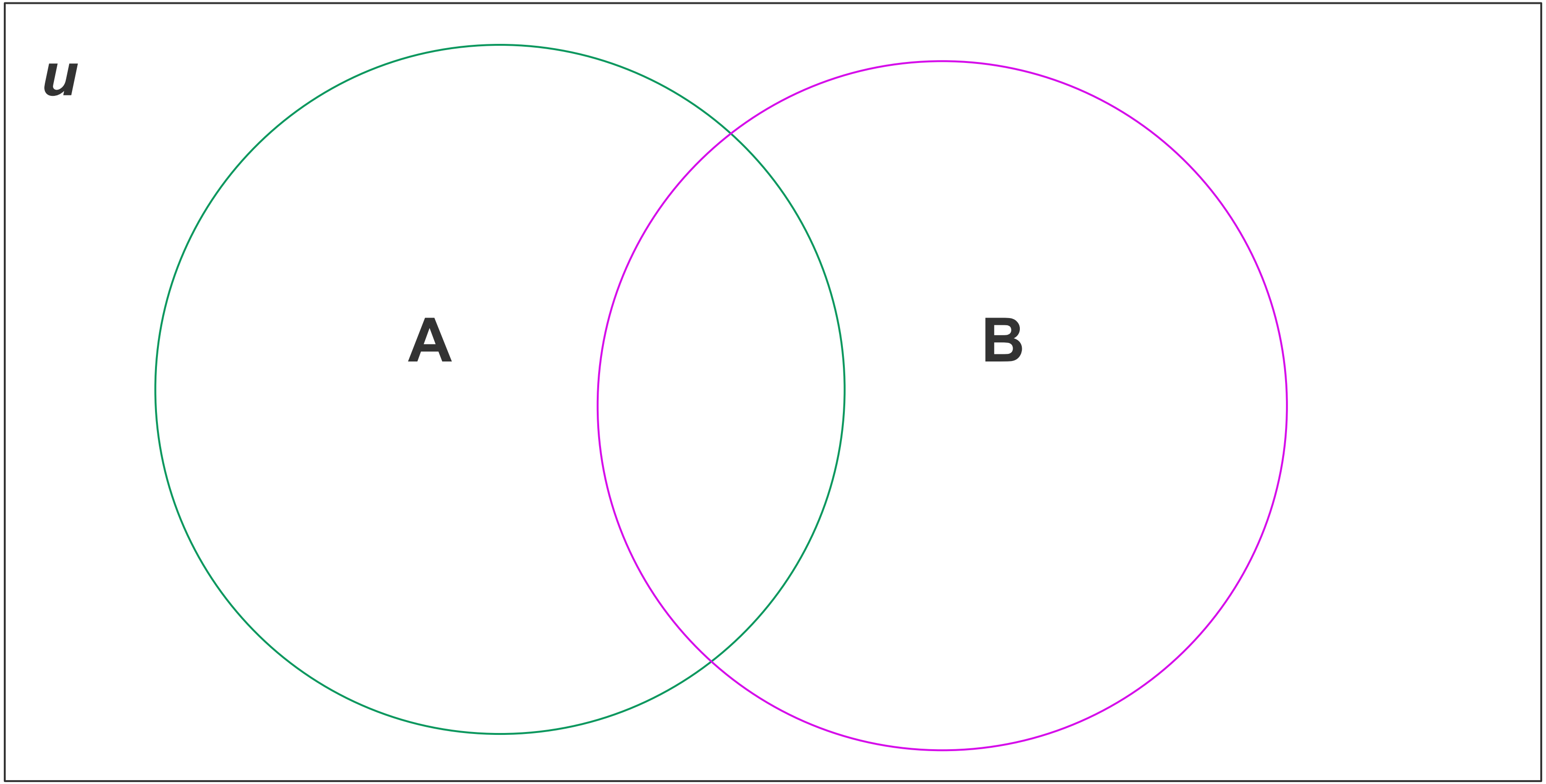
- A set is an **unordered** group of **distinct** elements
  - Set variable names are capital letters, with lower-case letters for elements

- Set Notation:
  - $a \in A$: "a is in A" or "a is an element of A"
  - $A \subseteq B$: "A is a subset of B", every element of A is also in B
  - $\emptyset$: "empty set", a unique set containing no elements
  - $\mathcal{P}(A)$: "power set of A", the set of all subsets of A including the empty set and A itself

Set Operators

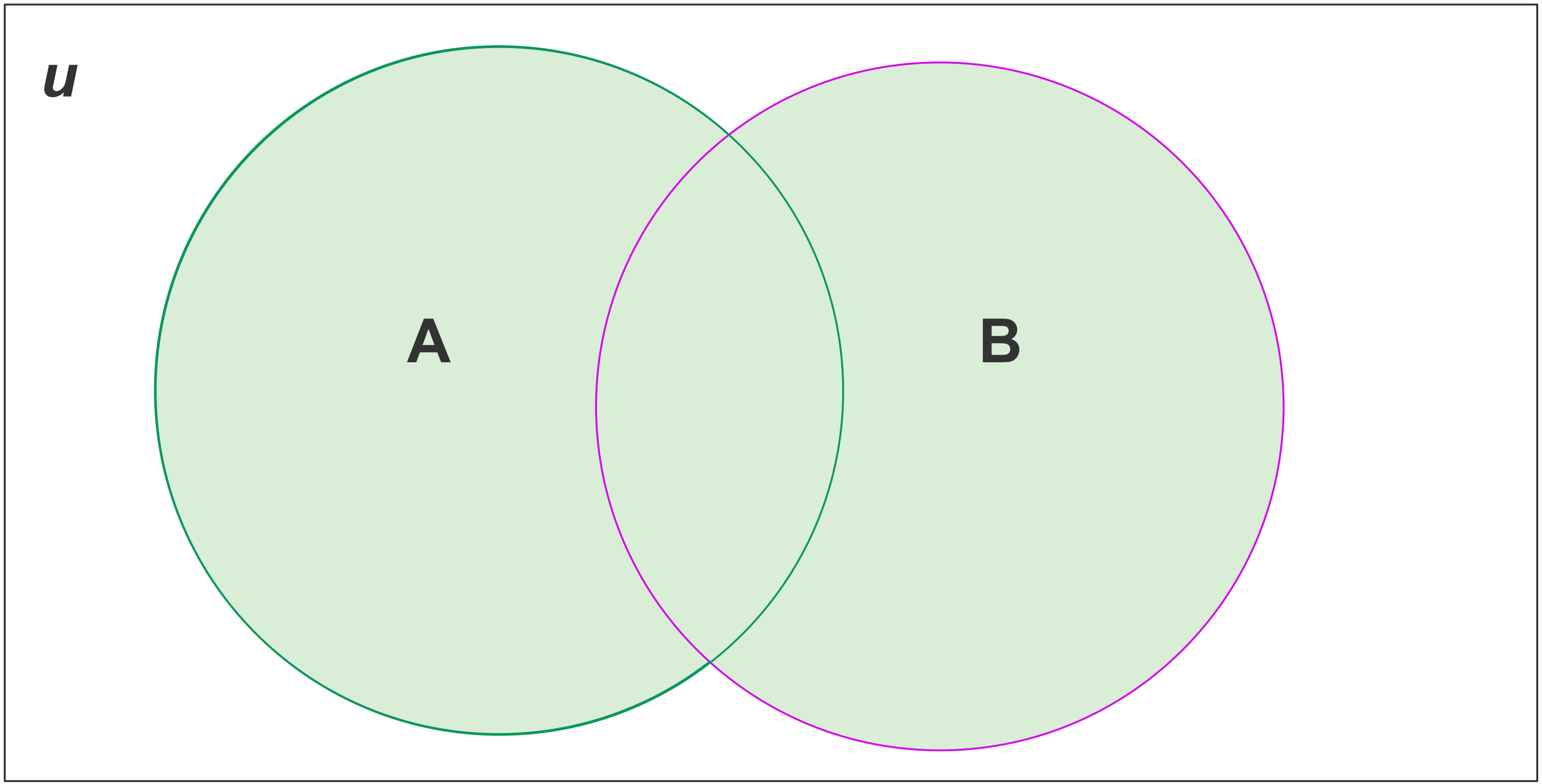
- Subset: $A \subseteq B \equiv \forall x(x \in A \rightarrow x \in B)$
- Equality: $A = B \equiv \forall x(x \in A \leftrightarrow x \in B) \equiv A \subseteq B \wedge B \subseteq A$
- Union: $A \cup B = \{x : x \in A \vee x \in B\}$
- Intersection: $A \cap B = \{x : x \in A \wedge x \in B\}$
- Complement: $\overline{A} = \{x : x \notin A\}$
- Difference: $A \backslash \mathrm{B} = \{x : x \in A \wedge x \notin B\}$
- Cartesian Product: $A \times B = \{(a, b) : a \in A \wedge b \in B\}$

Understand Sets Visually!



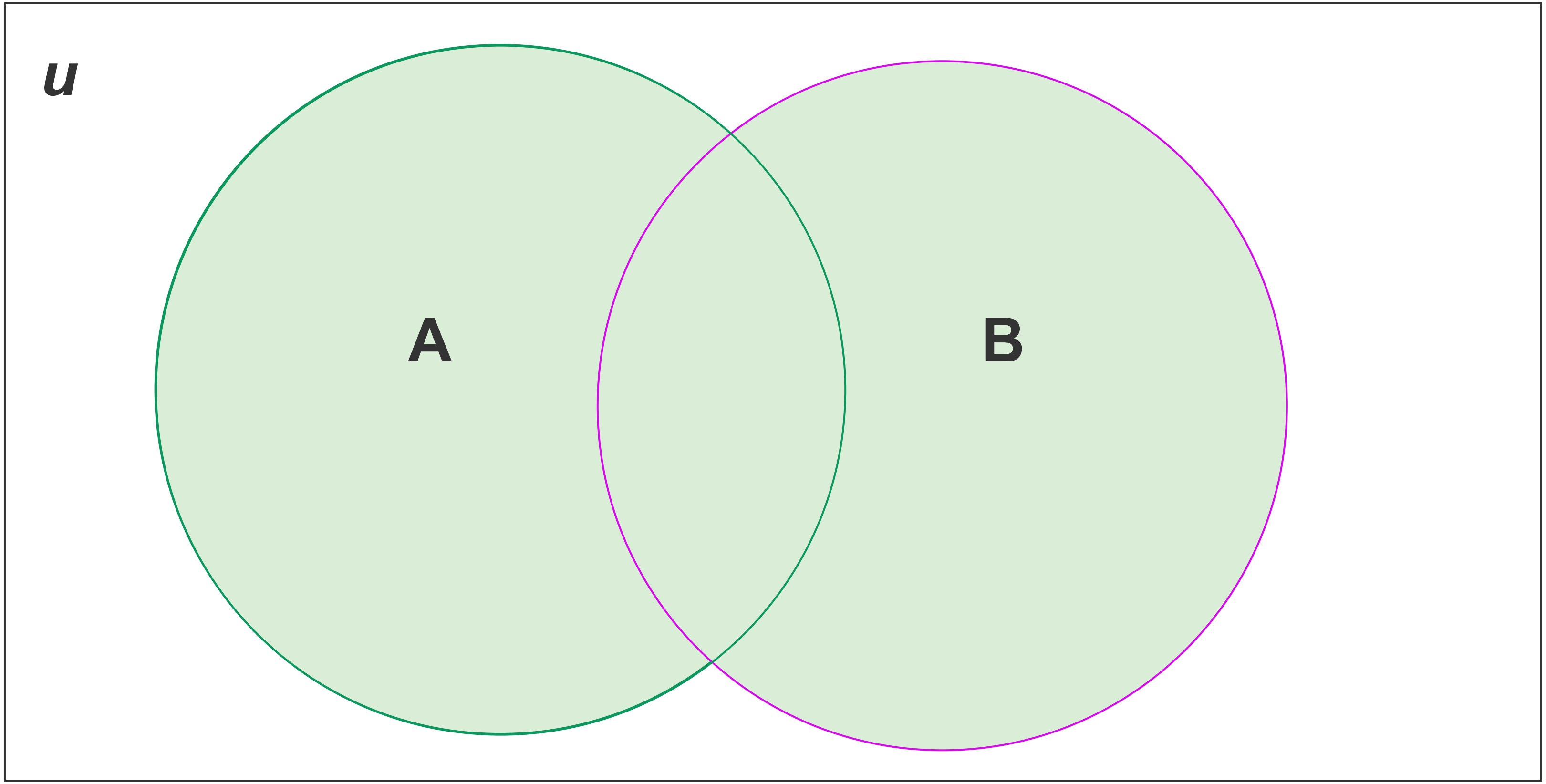
Understand Sets Visually!

What Set Operation is this?



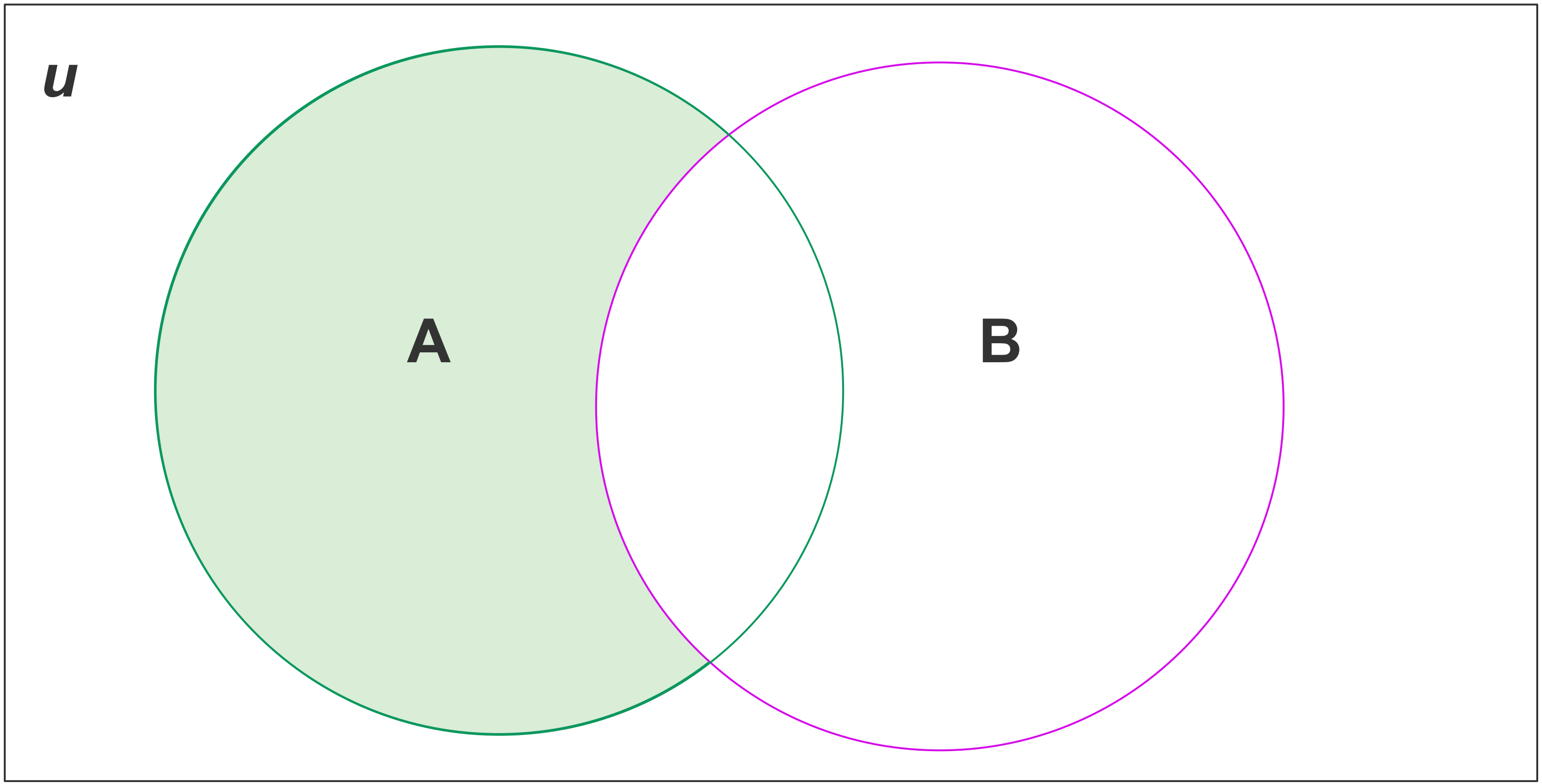
Understand Sets Visually!

Union: A U B



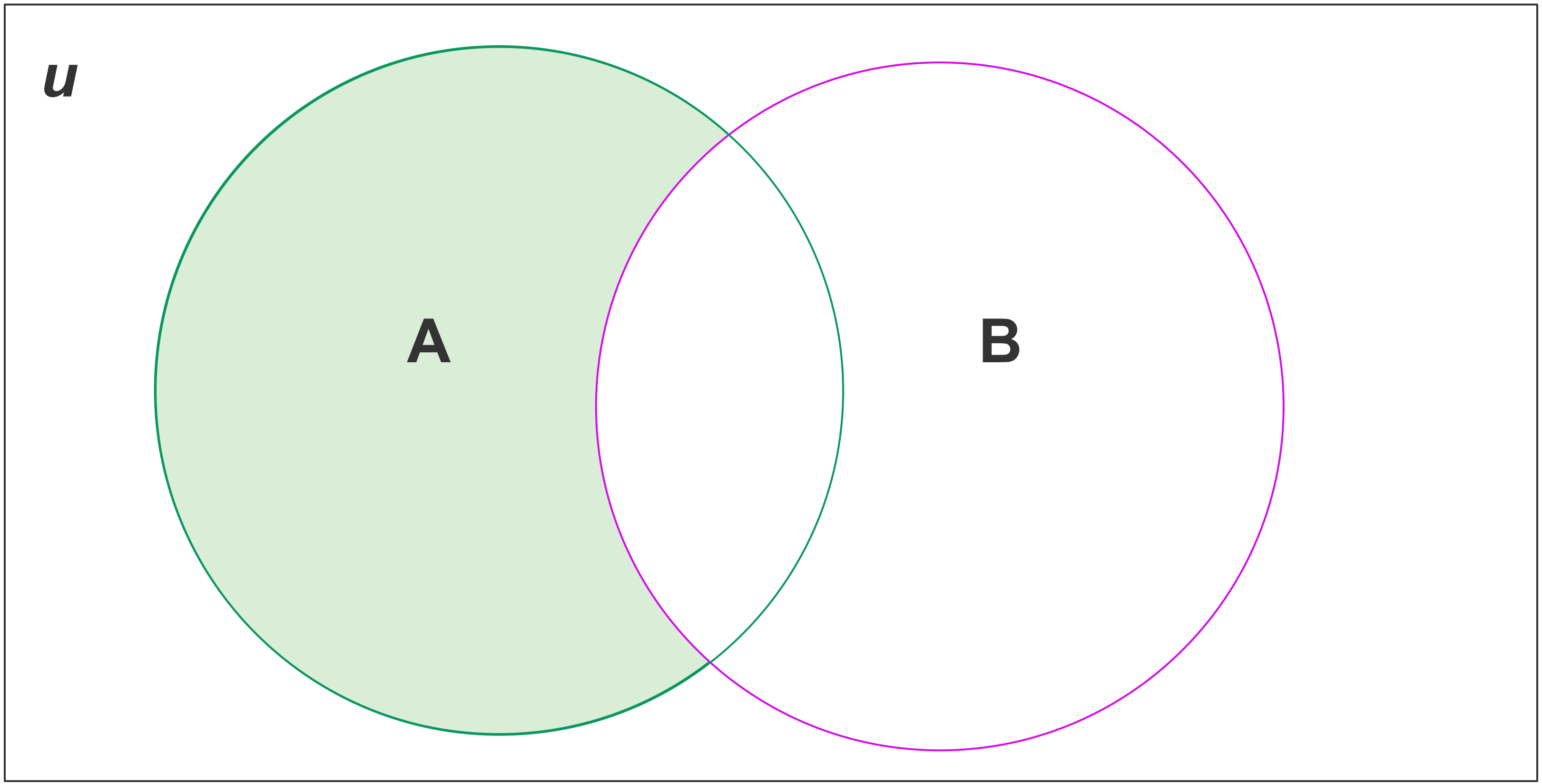
Understand Sets Visually!

What Set Operation is this?



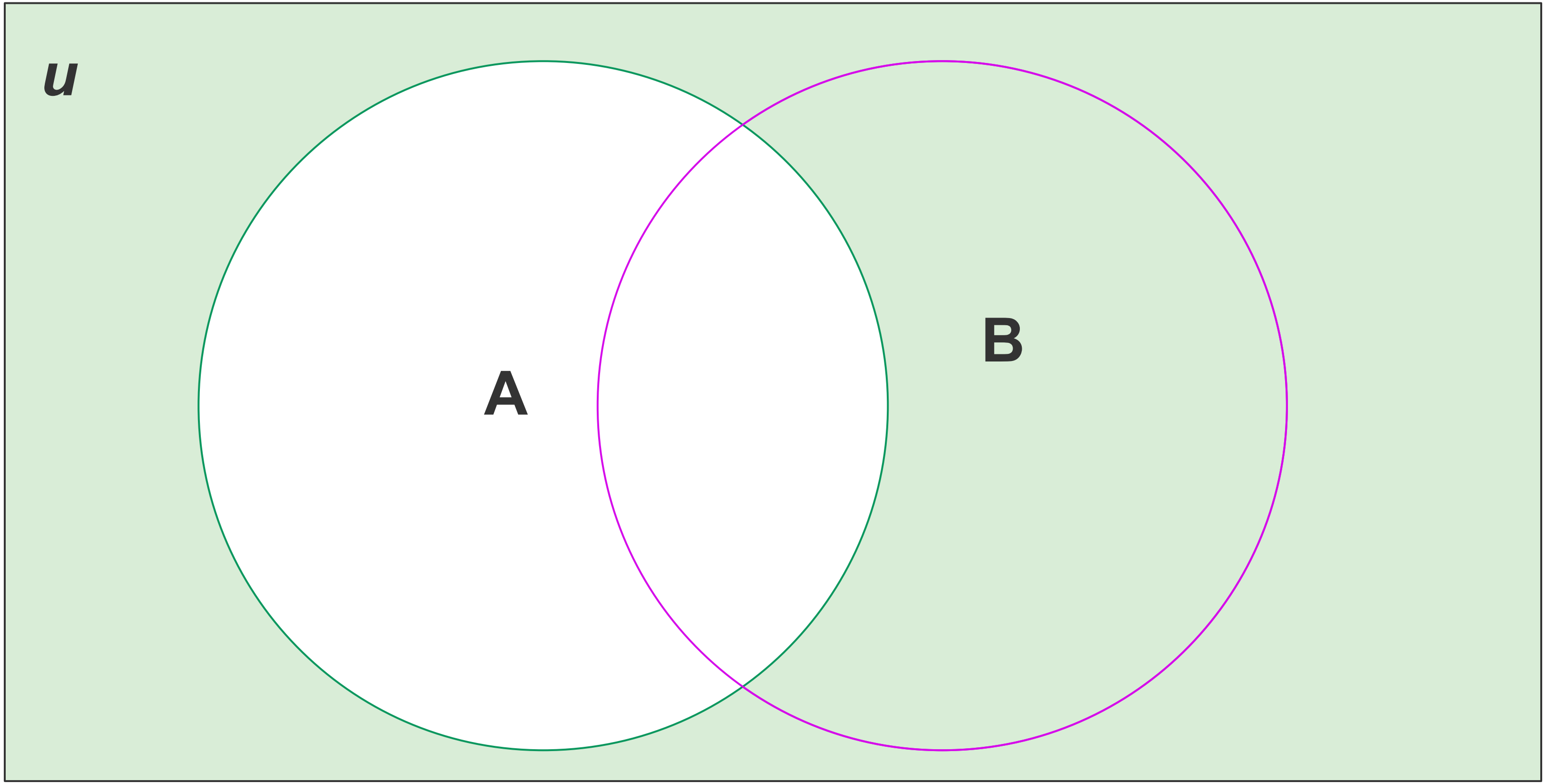
Understand Sets Visually!

Difference: A \ B



Understand Sets Visually!

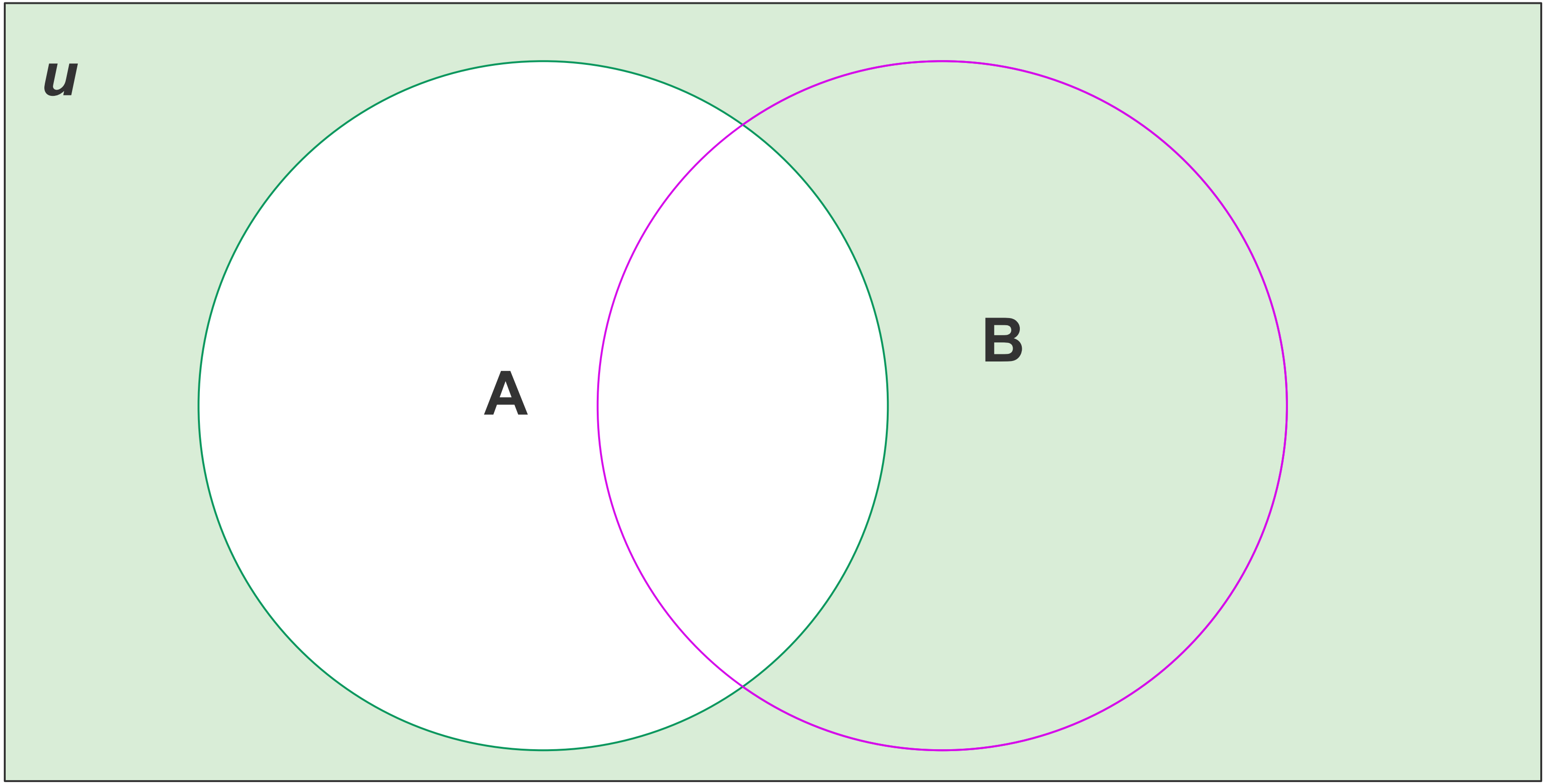
What Set Operation is this?



Understand Sets Visually!

A complement:

$\overline{A}$



Cartesian Product!

Problem 1: Cartesian Product

Definition of Cartesian Product:
$A \times B ::= \{x : \exists a \in A, \exists b \in B\ (x = (a, b))\}$

Let A, B, C, and D be sets. Consider the following claim:

$$(A \cap B) \times C \subseteq A \times (C \cup D)$$

a) Suppose that $A = \{1, 2\}$, $B = \{1, 2, 3\}, C = \{3, 4\}, D = \{2\}$.

Calculate the values of the sets $(A \cap B) \times C$ and $A \times (C \cup D)$. Check whether the claim holds.

Cartesian Product

Let A, B, C, and D be sets. Consider the following claim:

$$(A \cap B) \times C \subseteq A \times (C \cup D)$$

a) Suppose that $A = \{1, 2\}$, $B = \{1, 2, 3\}, C = \{3, 4\}, D = \{2\}$.

Calculate the values of the sets $(A \cap B) \times C$ and $A \times (C \cup D)$. Check whether the claim holds.

$$A \cap B = \{1, 2\}$$

Cartesian Product

Let A, B, C, and D be sets. Consider the following claim:

$$(A \cap B) \times C \subseteq A \times (C \cup D)$$

a) Suppose that $A = \{1, 2\}$, $B = \{1, 2, 3\}, C = \{3, 4\}, D = \{2\}$.

Calculate the values of the sets $(A \cap B) \times C$ and $A \times (C \cup D)$. Check whether the claim holds.

$$A \cap B = \{1, 2\}$$
$$(A \cap B) \times C = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$$

Cartesian Product

Let A, B, C, and D be sets. Consider the following claim:

$$(A \cap B) \times C \subseteq A \times (C \cup D)$$

a) Suppose that $A = \{1,2\}$, $B = \{1,2,3\}, C = \{3,4\}, D = \{2\}$.

Calculate the values of the sets $(A \cap B) \times C$ and $A \times (C \cup D)$. Check whether the claim holds.

$$A \cap B = \{1,2\}$$
$$(A \cap B) \times C = \{(1,3),(1,4),(2,3),(2,4)\}$$
$$C \cup D = \{2,3,4\}$$

Cartesian Product

Let A, B, C, and D be sets. Consider the following claim:

$$(A \cap B) \times C \subseteq A \times (C \cup D)$$

a) Suppose that $A = \{1, 2\}$, $B = \{1, 2, 3\}, C = \{3, 4\}, D = \{2\}$.

Calculate the values of the sets $(A \cap B) \times C$ and $A \times (C \cup D)$. Check whether the claim holds.

$$\begin{aligned} A \cap B &= \{1, 2\} \\ (A \cap B) \times C &= \{(1,3), (1,4), (2,3), (2,4)\} \\ C \cup D &= \{2, 3, 4\} \\ A \times (C \cup D) &= \{(1,2), (1,3), (1,4), (2,2), (2,3), (2,4)\} \end{aligned}$$

Cartesian Product

Let A, B, C, and D be sets. Consider the following claim:

$$(A \cap B) \times C \subseteq A \times (C \cup D)$$

a) Suppose that $A = \{1, 2\}$, $B = \{1, 2, 3\}, C = \{3, 4\}, D = \{2\}$.

Calculate the values of the sets $(A \cap B) \times C$ and $A \times (C \cup D)$. Check whether the claim holds.

$$A \cap B = \{1, 2\}$$
$$(A \cap B) \times C = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$$
$$C \cup D = \{2, 3, 4\}$$
$$A \times (C \cup D) = \{(1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4)\}$$

We can see that $(A \cap B) \times C \subseteq A \times (C \cup D)$. The claim holds.

Cartesian Product

$(A \cap B) \times C \subseteq A \times (C \cup D)$

c) Write an **English proof** that the claim holds.

Follow the structure of our template for subset proofs.

Note: even though we want you to write your proof directly in English, it must still look like the translation of a formal proof. In particular, you must include all steps that would be required of a formal proof, excepting only those that we have explicitly said are okay to skip in English proofs.

Cartesian Product

$(A \cap B) \times C \subseteq A \times (C \cup D)$

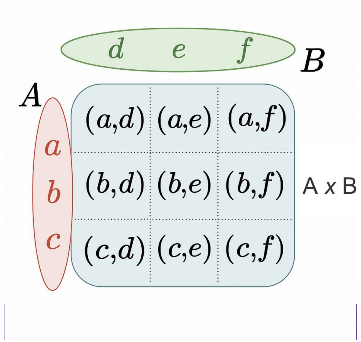
Let ***x*** be an **arbitrary** object.

Since ***x*** was **arbitrary**, we have shown that $(A \cap B) \times C \subseteq A \times (C \cup D)$ by the definition of subset.

Cartesian Product

(*A* ∩ *B*) x *C* ⊆ *A* x (*C* ∪ *D*)

Remember, **x** is just a coordinate point!



Let ***x*** be an **arbitrary** object.

Suppose that *x* ∈ (*A* ∩ *B*) x *C*.

Since ***x*** was **arbitrary**, we have shown that (*A* ∩ *B*) x *C* ⊆ *A* x (*C* ∪ *D*) by the definition of subset.

Cartesian Product

$(A \cap B) \times C \subseteq A \times (C \cup D)$

Let ***x*** be an **arbitrary** object.

Suppose that $x \in (A \cap B) \times C$

Then, by definition of Cartesian product, there is some $y \in A \cap B$ and $c \in C$ such that $x = (y, c)$.

Since ***x*** was **arbitrary**, we have shown that $(A \cap B) \times C \subseteq A \times (C \cup D)$ by the definition of subset.

Cartesian Product

$(A \cap B) \times C \subseteq A \times (C \cup D)$

Let ***x*** be an **arbitrary** object.

Suppose that $x \in (A \cap B) \times C$

Then, by definition of Cartesian product, there is some $y \in A \cap B$ and $c \in C$ such that $x = (y, c)$.

Then, by the definition of intersection, $y \in A$ and $y \in B$.

Since ***x*** was **arbitrary**, we have shown that $(A \cap B) \times C \subseteq A \times (C \cup D)$ by the definition of subset.

Cartesian Product

$(A \cap B) \times C \subseteq A \times (C \cup D)$

Work a step back!

Let ***x*** be an **arbitrary** object.

Suppose that $x \in (A \cap B) \times C$

Then, by definition of Cartesian product, there is some $y \in A \cap B$ and $c \in C$ such that $x = (y, c)$.

Then, by the definition of intersection, $y \in A$ and $y \in B$.

Since $y \in A$ and $c \in C \cup D$, we can see that $x = (y, c) \in A \times (C \cup D)$ by the definition of Cartesian product.

Since ***x*** was **arbitrary**, we have shown that $(A \cap B) \times C \subseteq A \times (C \cup D)$ by the definition of subset.

Cartesian Product

$(A \cap B) \times C \subseteq A \times (C \cup D)$

Let ***x*** be an **arbitrary** object.

Suppose that $x \in (A \cap B) \times C$

Then, by definition of Cartesian product, there is some $y \in A \cap B$ and $c \in C$ such that $x = (y, c)$.

Then, by the definition of intersection, $y \in A$ and $y \in B$.

Since $c \in C$, then $c \in C$ or $c \in D$. Then, by definition of union, $c \in C \cup D$.

Since $y \in A$ and $c \in C \cup D$, we can see that $x = (y, c) \in A \times (C \cup D)$ by the definition of Cartesian product.

Since ***x*** was **arbitrary**, we have shown that $(A \cap B) \times C \subseteq A \times (C \cup D)$ by the definition of subset.

Structural Induction

Idea of Structural Induction

Every **element is built up recursively…**

So to show $P(s)$ for all s…

Show $P(b)$ for all base case elements b.

Show for an arbitrary element not in the base case, if $P()$ holds for every named element in the recursive rule, then $P()$ holds for the new element (each recursive rule will be a case of this proof).

Structural Induction Template

Let $P(x)$ be “<predicate>”. We show $P(x)$ holds for all x by structural induction.

Base Case: Show $P(x)$
[Do that for every base cases x.]

Inductive Hypothesis: Suppose $P(x)$ for an arbitrary x
[Do that for every x listed as in S in the recursive rules.]

Inductive Step: Show $P()$ holds for y.
[You will need a separate case/step for every recursive rule.]

Therefore $P(x)$ holds for all x by the principle of induction.

Problem 4 - Structural Induction

Let P(L) be “”.
We show P(L) holds for all … L by structural induction on L.

Base Case: Show P(L) (for all L in the basis rules)

Inductive Hypothesis: Suppose P(L) (for all L in the recursive rules),
i.e. (IH in terms of P(L))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(L) holds for all L ∈ S by the principle of induction.

Problem 4 - Structural Induction

Let P(L) be “len(echo-pos(L)) ≤ 2 len(L)”.
We show P(L) holds for all lists L by structural induction on L.

Base Case: Show P(L) (for all L in the basis rules)

Inductive Hypothesis: Suppose P(L) (for all L in the recursive rules),
i.e. (IH in terms of P(L))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(L) holds for all L ∈ S by the principle of induction.

Problem 4 - Structural Induction

Let P(L) be “len(echo-pos(L)) ≤ 2 len(L)”.
We show P(L) holds for all lists L by structural induction on L.

Base Case: Show P(nil)

len(echo-pos(nil)) ≤ len(nil)

So P(nil) holds.

Inductive Hypothesis: Suppose P(L) (for all L in the recursive rules),
i.e. (IH in terms of P(L))

Inductive Step: Goal: Show that P(?) holds. (IS goal in terms of P(?))

Conclusion: Therefore P(L) holds for all L ∈ S by the principle of induction.

Problem 4 - Structural Induction

Let P(L) be "len(echo-pos(L)) ≤ 2 len(L)".
We show P(L) holds for all lists L by structural induction on L.

<u>Base Case:</u> Show P(nil)

len(echo-pos(nil)) ≤ len(nil)

So P(nil) holds.

<u>Inductive Hypothesis:</u> Suppose P(L) holds for an arbitrary L ∈ List

<u>Inductive Step:</u> Goal: Show that P(?) holds. (IS goal in terms of P(?))

<u>Conclusion:</u> Therefore P(L) holds for all L ∈ S by the principle of induction.

Problem 4 - Structural Induction

Let P(L) be "len(echo-pos(L)) ≤ 2 len(L)".
We show P(L) holds for all lists L by structural induction on L.

Base Case: Show P(nil)

len(echo-pos(nil)) ≤ len(nil)

So P(nil) holds.

Inductive Hypothesis: Suppose P(L) holds for an arbitrary L ∈ List

Inductive Step: Goal: Show that P(a::L) holds.

Conclusion: Therefore P(L) holds for all L ∈ S by the principle of induction.

Problem 4 - Structural Induction

Let P(L) be “len(echo-pos(L)) ≤ 2 len(L)”.
We show P(L) holds for all lists L by structural induction on L.

Base Case: Show P(nil)

    len(echo-pos(nil)) ≤ len(nil)

So P(nil) holds.

Inductive Hypothesis: Suppose P(L) holds for an arbitrary L ∈ List

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Conclusion: Therefore P(L) holds for all L ∈ S by the principle of induction.

Problem 4 - Structural Induction

Let P(L) be “len(echo-pos(L)) ≤ 2 len(L)”.
We show P(L) holds for all lists L by structural induction on L.

Base Case: Show P(nil)

len(echo-pos(nil)) ≤ len(nil)

So P(nil) holds.

Inductive Hypothesis: Suppose P(L) holds for an arbitrary L ∈ List

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.
Suppose that a ≤ 0

We can do casework!

Suppose that a > 0

Conclusion: Therefore P(L) holds for all L ∈ S by the principle of induction.

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.
Suppose that a ≤ 0

Suppose that a > 0

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.
Suppose that a ≤ 0

len(echo-pos(a::L)) =

Suppose that a > 0

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) && \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) && \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L)) = len(a::echo-pos(L)) Def of echo-pos (since a ≤ 0)

Suppose that a > 0

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len

Suppose that a > 0

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) && \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) && \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH

Suppose that a > 0

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

<u>Inductive Step:</u> Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	

Suppose that a > 0

$$\begin{aligned} \mathsf{len}(\mathsf{nil}) &:= 0 \\ \mathsf{len}(a :: L) &:= 1 + \mathsf{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \mathsf{echo\text{-}pos}(\mathsf{nil}) &:= \mathsf{nil} \\ \mathsf{echo\text{-}pos}(a :: L) &:= a :: \mathsf{echo\text{-}pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \mathsf{echo\text{-}pos}(a :: L) &:= a :: a :: \mathsf{echo\text{-}pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

<u>Inductive Step:</u> Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	

Suppose that a > 0

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

<u>Inductive Step:</u> Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

len(echo-pos(a::L)) =

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

len(echo-pos(a::L))	= len(a::a::echo-pos(L))	Def of echo-pos (since a > 0)

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

<u>Inductive Step:</u> Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

len(echo-pos(a::L))	= len(a::a::echo-pos(L))	Def of echo-pos (since a > 0)
	= 1 + len(a::echo-pos(L))	Def of len

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

<u>Inductive Step:</u> Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

len(echo-pos(a::L))	= len(a::a::echo-pos(L))	Def of echo-pos (since a > 0)
	= 1 + len(a::echo-pos(L))	Def of len
	= 1 + 1 + len(echo-pos(L))	Def of len

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) \quad \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) \quad \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

Problem 4 - Structural Induction

<u>Inductive Step:</u> Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

len(echo-pos(a::L))	= len(a::a::echo-pos(L))	Def of echo-pos (since a > 0)
	= 1 + len(a::echo-pos(L))	Def of len
	= 1 + 1 + len(echo-pos(L))	Def of len
	≤ 2 + 2len(L)	IH

Problem 4 - Structural Induction

<u>Inductive Step:</u> Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

len(echo-pos(a::L))	= len(a::a::echo-pos(L))	Def of echo-pos (since a > 0)
	= 1 + len(a::echo-pos(L))	Def of len
	= 1 + 1 + len(echo-pos(L))	Def of len
	≤ 2 + 2len(L)	IH
	= 2(1 + len(L))	

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

len(echo-pos(a::L))	= len(a::a::echo-pos(L))	Def of echo-pos (since a > 0)
	= 1 + len(a::echo-pos(L))	Def of len
	= 1 + 1 + len(echo-pos(L))	Def of len
	≤ 2 + 2len(L)	IH
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Problem 4 - Structural Induction

Inductive Step: Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

len(echo-pos(a::L))	= len(a::a::echo-pos(L))	Def of echo-pos (since a > 0)
	= 1 + len(a::echo-pos(L))	Def of len
	= 1 + 1 + len(echo-pos(L))	Def of len
	≤ 2 + 2len(L)	IH
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Since these cases are exhaustive, P(a::L) holds.

Problem 4 - Structural Induction

<u>Inductive Step:</u> Goal: Show that P(a::L) holds. Let a ∈ Z be arbitrary.

Suppose that a ≤ 0

len(echo-pos(a::L))	= len(a::echo-pos(L))	Def of echo-pos (since a ≤ 0)
	= 1 + len(echo-pos(L))	Def of len
	≤ 1 + 2len(L)	IH
	≤ 2 + 2len(L)	
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Suppose that a > 0

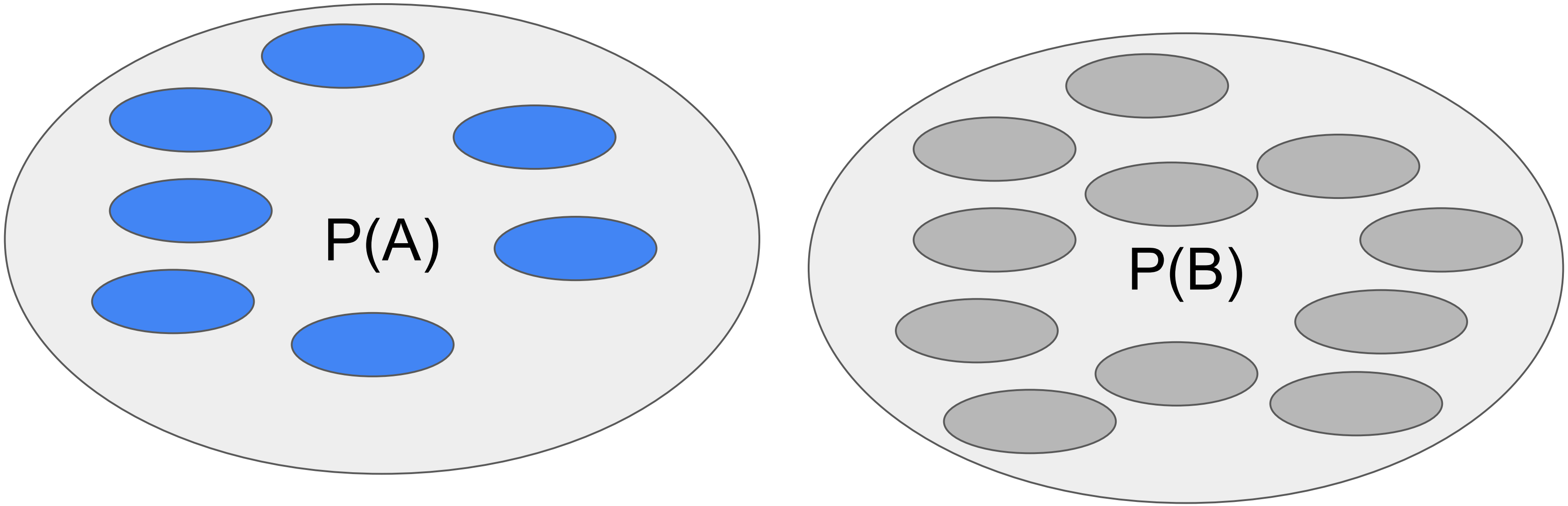
len(echo-pos(a::L))	= len(a::a::echo-pos(L))	Def of echo-pos (since a > 0)
	= 1 + len(a::echo-pos(L))	Def of len
	= 1 + 1 + len(echo-pos(L))	Def of len
	≤ 2 + 2len(L)	IH
	= 2(1 + len(L))	
	= 2(len(a::L))	Def of len

Since these cases are exhaustive, P(a::L) holds.

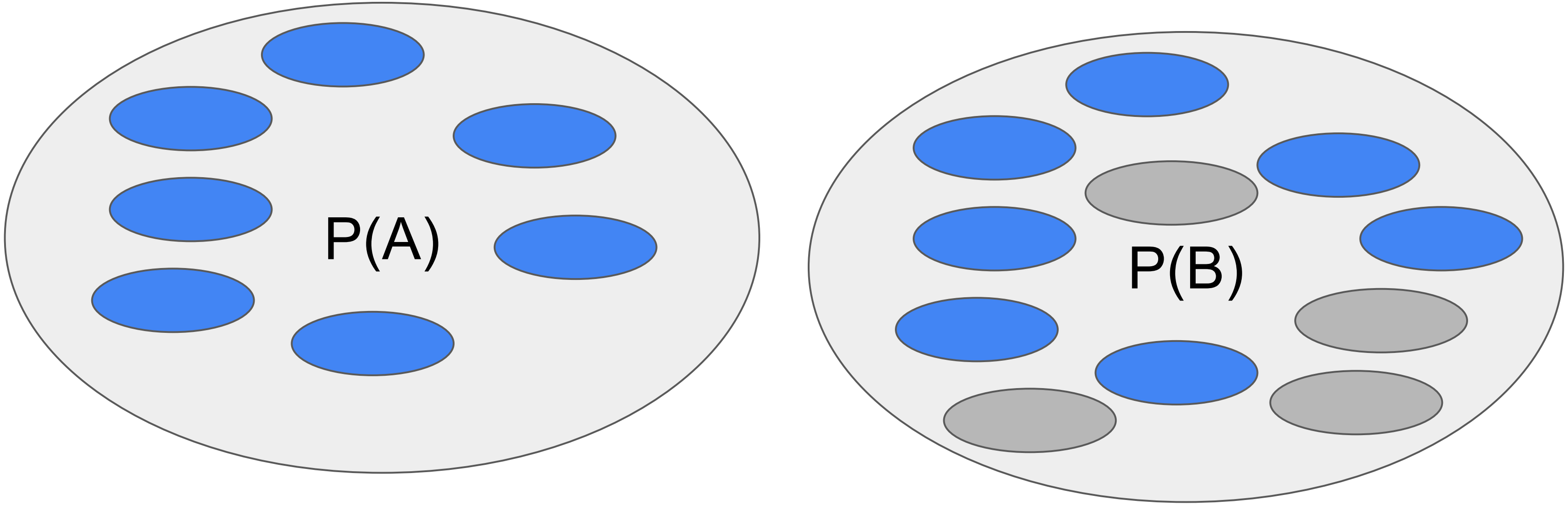
<u>Conclusion:</u> Therefore P(L) holds for all L ∈ List by the principle of induction.

Powerset English Proof

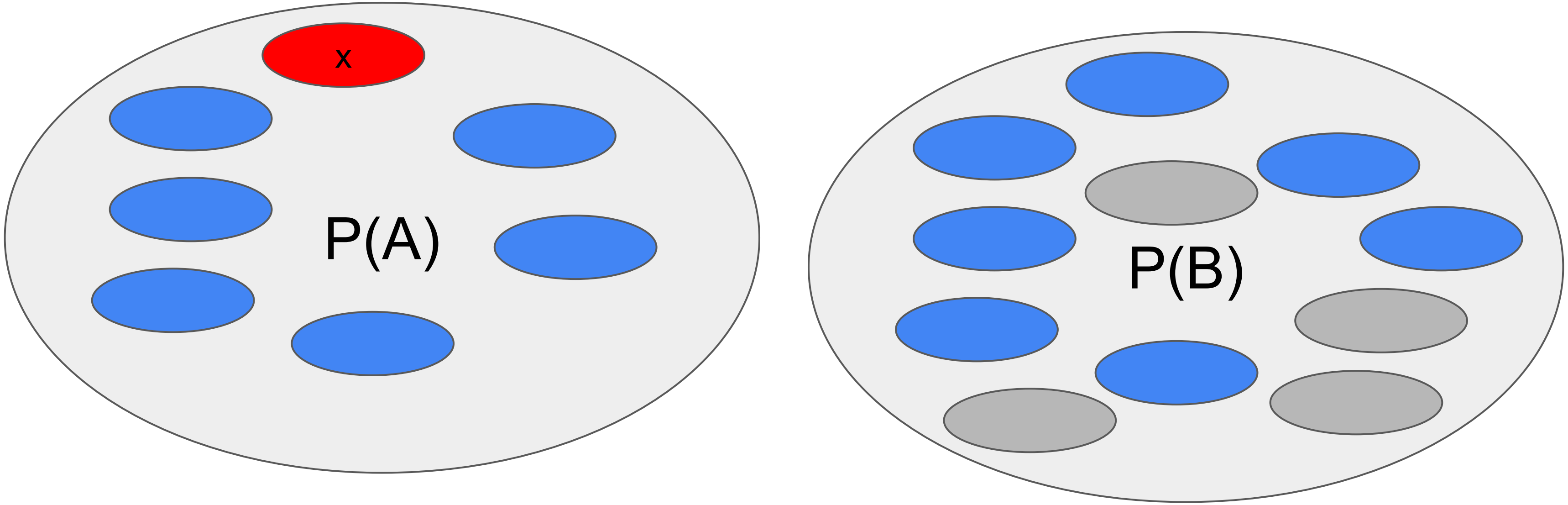
Given $A \subseteq B$, we want $P(A) \subseteq P(B)$



To show P(A) ⊆ P(B), show that the (set) elements of P(A) can be found in P(B)

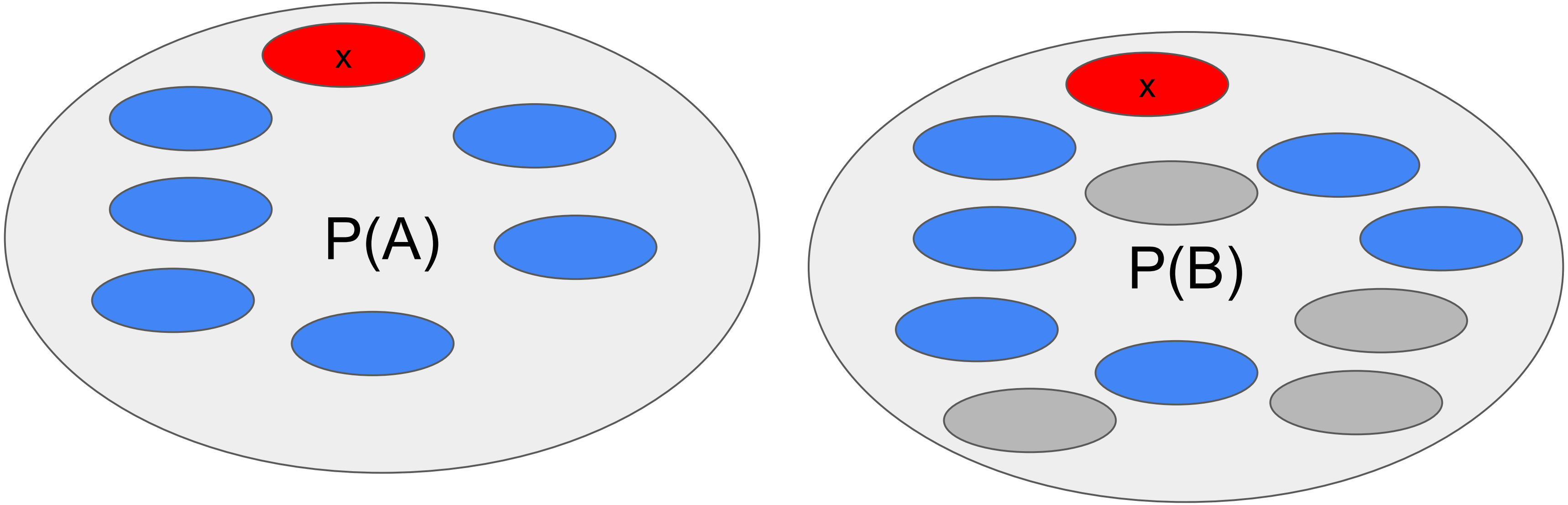


Subset proof strategy: take an arbitrary element x of P(A)...

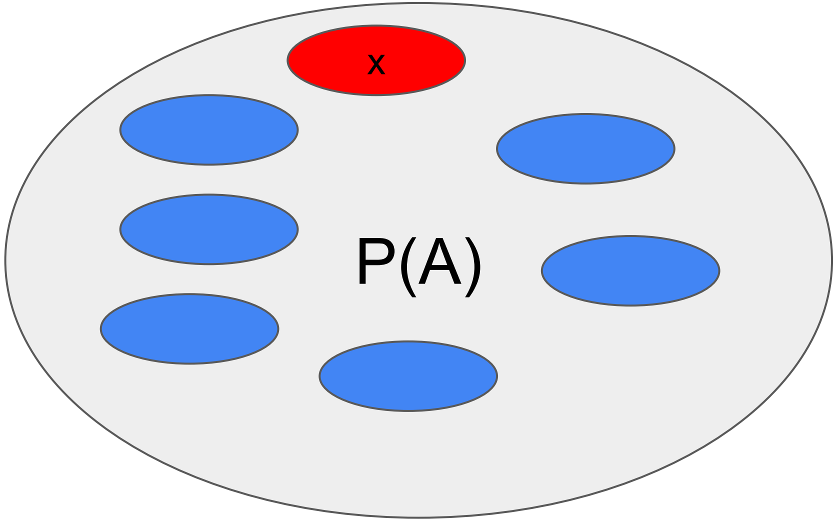


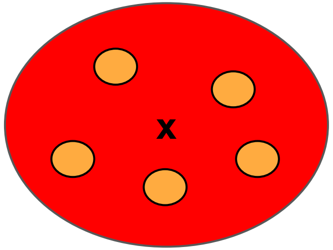
Subset proof strategy: … and show that it's in P(B)

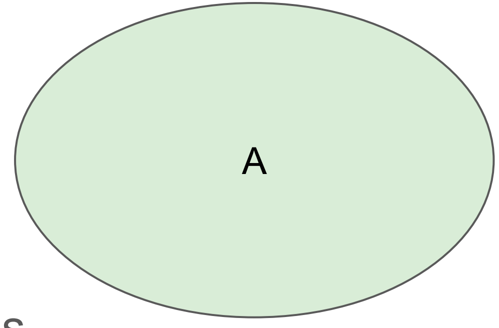
How do we show x is in P(B)?



Well, x is in P(A), so x $\subseteq$ A by definition of powerset. Our target is showing x is in P(B), i.e., x $\subseteq$ B.

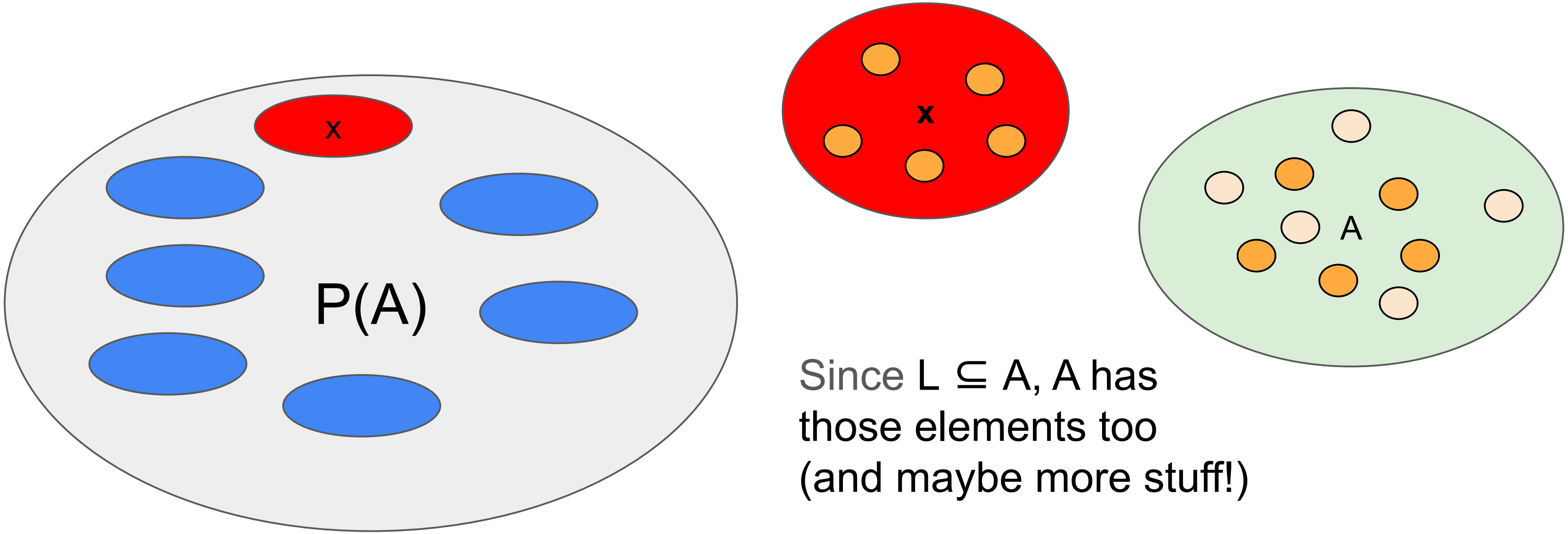






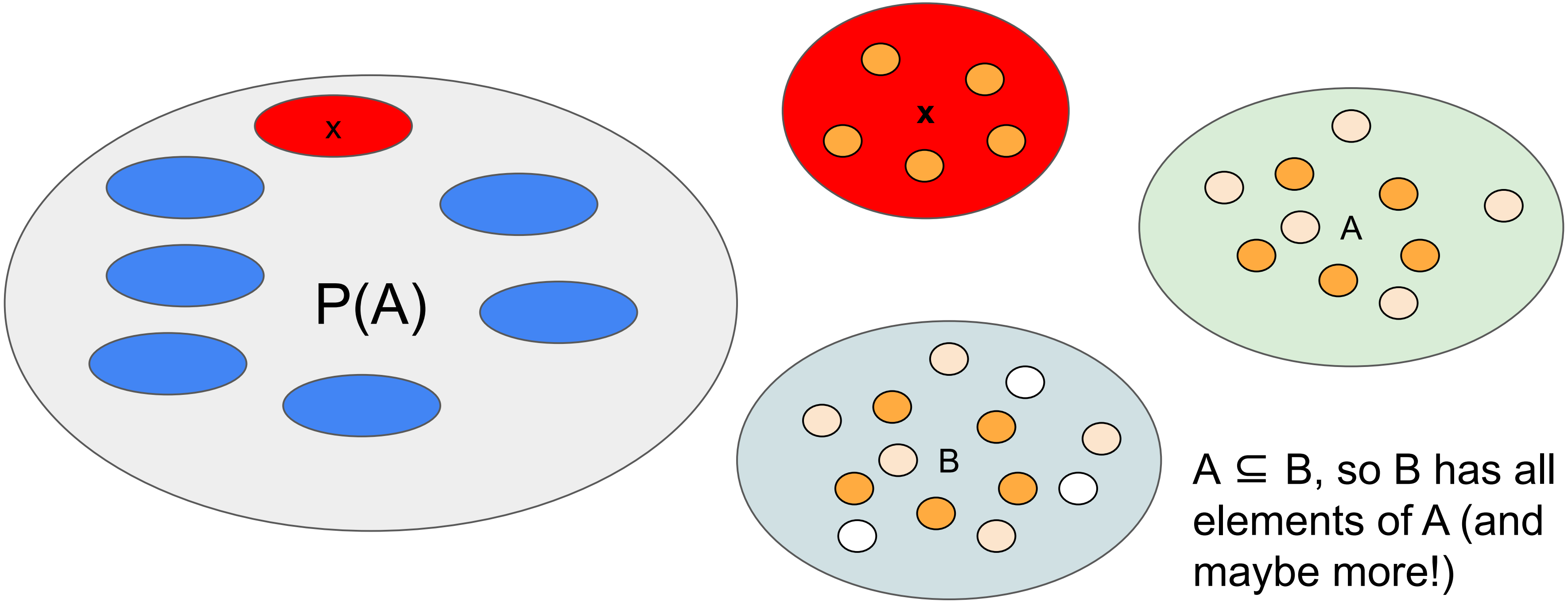
L is a set, so it has elements in it!

Well, x is in P(A), so x ⊆ A. Our target is showing x is in P(B), i.e., x ⊆ B.



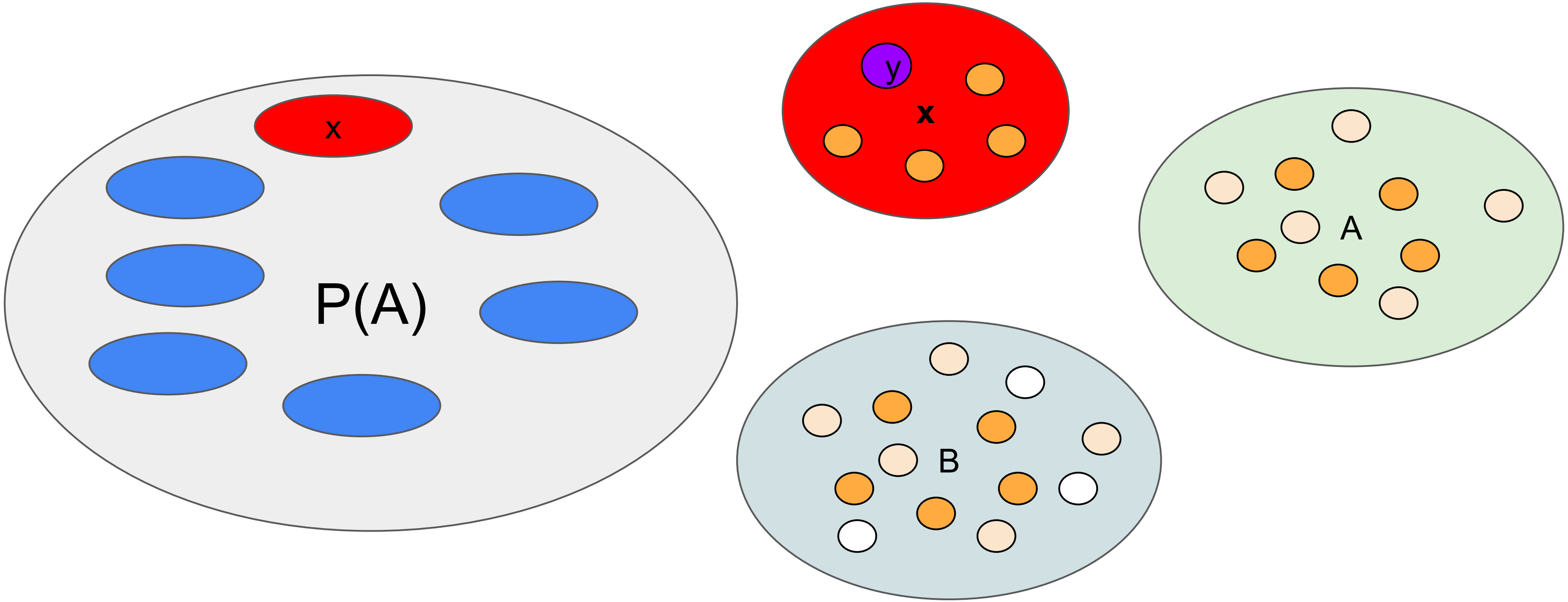
Since L ⊆ A, A has those elements too (and maybe more stuff!)

Well, x is in P(A), so x ⊆ A. Our target is showing x is in P(B), i.e., x ⊆ B.

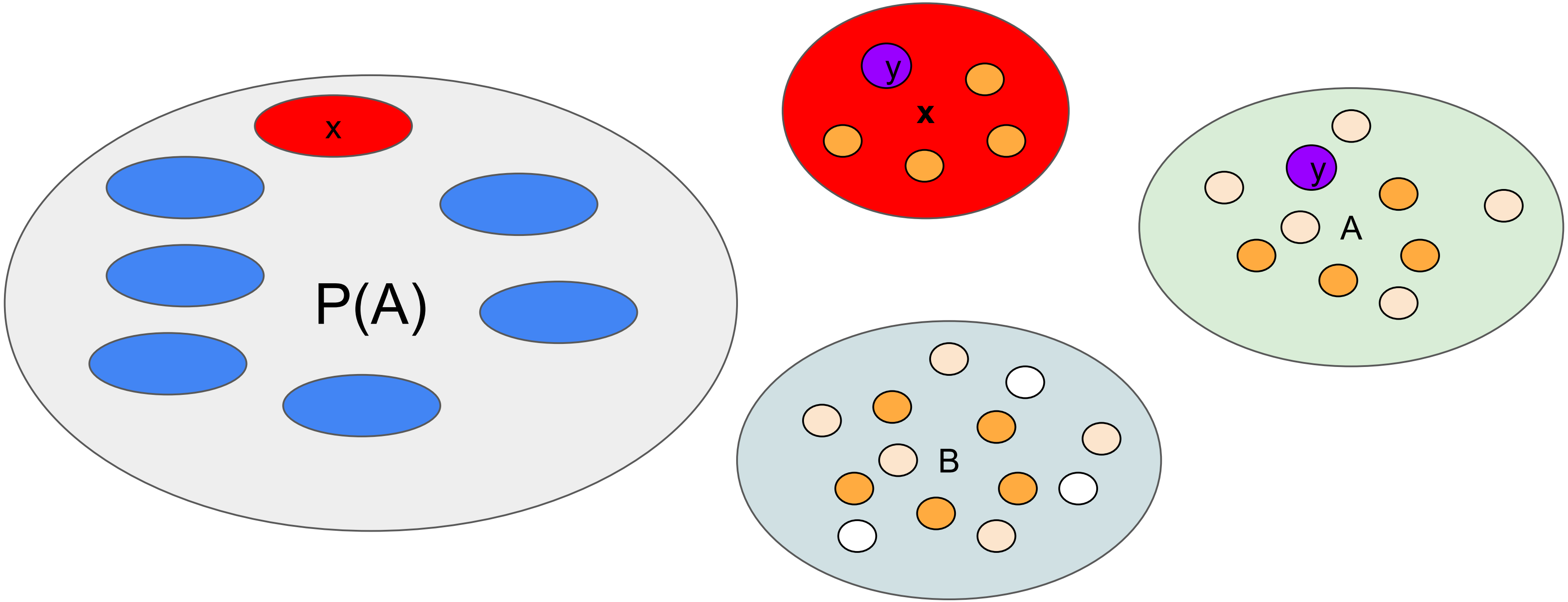


A ⊆ B, so B has all the elements of A (and maybe more!)

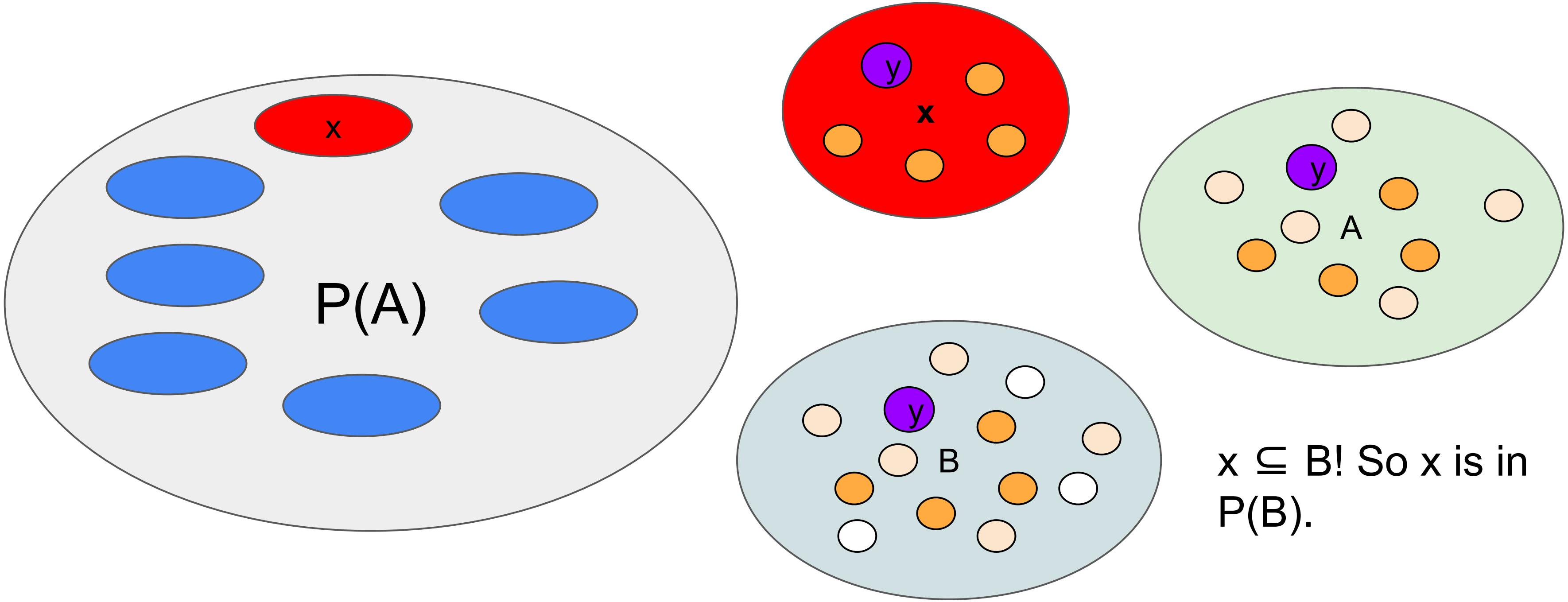
To show x ⊆ B, we do the subset strategy again: take an arbitrary y in x…



To show x ⊆ B, we do the subset strategy again: Since x ⊆ A, y is in A…



To show x ⊆ B, we do the subset strategy again: And finally since A ⊆ B, y is in B.



x ⊆ B! So x is in P(B).

Problem 2c

Let A and B be sets. Prove that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ follows from $A \subseteq B$.

Problem 2c

Let A and B be sets. Prove that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ follows from $A \subseteq B$.

Let X be an arbitrary set in $\mathcal{P}(A)$.

Problem 2c

Let A and B be sets. Prove that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ follows from $A \subseteq B$.

Let X be an arbitrary set in $\mathcal{P}(A)$. By definition of power set, $X \subseteq A$. We need to show that $X \in \mathcal{P}(B)$, or equivalently, that $X \subseteq B$.

Problem 2c

Let A and B be sets. Prove that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ follows from $A \subseteq B$.

Let X be an arbitrary set in $\mathcal{P}(A)$. By definition of power set, $X \subseteq A$. We need to show that $X \in \mathcal{P}(B)$, or equivalently, that $X \subseteq B$.

Since X was an arbitrary set, any set in $\mathcal{P}(A)$ is in $\mathcal{P}(B)$, or, by definition of subset, $\mathcal{P}(A) \subseteq \mathcal{P}(B)$. We have shown the claim.

Problem 2c

Let A and B be sets. Prove that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ follows from $A \subseteq B$.

Let X be an arbitrary set in $\mathcal{P}(A)$. By definition of power set, $X \subseteq A$. We need to show that $X \in \mathcal{P}(B)$, or equivalently, that $X \subseteq B$.

Let x be an arbitrary element of X.

Since X was an arbitrary set, any set in $\mathcal{P}(A)$ is in $\mathcal{P}(B)$, or, by definition of subset, $\mathcal{P}(A) \subseteq \mathcal{P}(B)$. We have shown the claim.

Problem 2c

Let A and B be sets. Prove that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ follows from $A \subseteq B$.

Let X be an arbitrary set in $\mathcal{P}(A)$. By definition of power set, $X \subseteq A$. We need to show that $X \in \mathcal{P}(B)$, or equivalently, that $X \subseteq B$.

Let x be an arbitrary element of X. Since $X \subseteq A$, it must be the case that $x \in A$. We were given that $A \subseteq B$. By definition of subset, any element of A is an element of B. So, it must also be the case that $x \in B$.

Since X was an arbitrary set, any set in $\mathcal{P}(A)$ is in $\mathcal{P}(B)$, or, by definition of subset, $\mathcal{P}(A) \subseteq \mathcal{P}(B)$. We have shown the claim.

Problem 2c

Let A and B be sets. Prove that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ follows from $A \subseteq B$.

Let X be an arbitrary set in $\mathcal{P}(A)$. By definition of power set, $X \subseteq A$. We need to show that $X \in \mathcal{P}(B)$, or equivalently, that $X \subseteq B$.

Let x be an arbitrary element of X. Since $X \subseteq A$, it must be the case that $x \in A$. We were given that $A \subseteq B$. By definition of subset, any element of A is an element of B. So, it must also be the case that $x \in B$.

Since x was arbitrary, we know any element of X is an element of B.

Since X was an arbitrary set, any set in $\mathcal{P}(A)$ is in $\mathcal{P}(B)$, or, by definition of subset, $\mathcal{P}(A) \subseteq \mathcal{P}(B)$. We have shown the claim.

Problem 2c

Let A and B be sets. Prove that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$ follows from $A \subseteq B$.

Let X be an arbitrary set in $\mathcal{P}(A)$. By definition of power set, $X \subseteq A$. We need to show that $X \in \mathcal{P}(B)$, or equivalently, that $X \subseteq B$.

Let x be an arbitrary element of X. Since $X \subseteq A$, it must be the case that $x \in A$. We were given that $A \subseteq B$. By definition of subset, any element of A is an element of B. So, it must also be the case that $x \in B$.

Since x was arbitrary, we know any element of X is an element of B. By definition of subset, $X \subseteq B$. By definition of power set, $X \in \mathcal{P}(B)$.

Since X was an arbitrary set, any set in $\mathcal{P}(A)$ is in $\mathcal{P}(B)$, or, by definition of subset, $\mathcal{P}(A) \subseteq \mathcal{P}(B)$. We have shown the claim.

Problem 2c - Formal Proof

1.	$A \subseteq B$	Given
2.	$\forall z\, (z \in A \rightarrow z \in B)$	Def of subset (1)
	Let X be arbitrary	
	3.1.1 $X \in P(A)$	Assumption
	3.1.2 $X \subseteq A$	Def of powerset (3.1.1)
	3.1.3 $\forall y\, (y \in X \rightarrow y \in A)$	Def of subset (3.1.2)
	Let x be arbitrary	
	3.1.4.1 $x \in X \rightarrow x \in A$	Elim $\forall$ (3.1.3)
	3.1.4.2 $x \in A \rightarrow x \in B$	Elim $\forall$ (2)
	3.1.4.3.1 $x \in X$	Assumption
	3.1.4.3.2 $x \in A$	MP (3.1.4.3.1, 3.1.4.1)
	3.1.4.3.3 $x \in B$	MP (3.1.4.3.2, 3.1.4.2)
	3.1.4.3 $x \in X \rightarrow x \in B$	Direct Proof
	3.1.4 $\forall y\, (y \in X \rightarrow y \in B)$	Intro $\forall$ (3.1.4.3)
	3.1.5 $X \subseteq B$	Def of subset (3.1.4)
	3.1.6 $X \in P(B)$	Def of powerset (3.1.5)
	3.1 $X \in P(A) \rightarrow X \in P(B)$	Direct Proof
3.	$\forall Y\, (Y \in P(A) \rightarrow Y \in P(B))$	Intro $\forall$
4.	$P(A) \subseteq P(B)$	Def of subset (3)

Set Equality and Meta Theorem

Set Equality

- To prove that sets A, B are equal (A = B), you must prove that $A \subseteq B$ and $B \subseteq A$
- This can be done via two subset proofs or using Meta Theorem
- Meta Theorem proof structure:

"Proof": Let x be an arbitrary object.

The stated bi-condition holds since:

$x \in$ left side $\equiv$ replace set ops with propositional logic

$\equiv$ apply Propositional Logic equivalence

$\equiv$ replace propositional logic with set ops

$\equiv x \in$ right side

Since x was arbitrary, we have shown, by definition, that the sets are equal. ■

Set Equality

Some equivalences you can use in a Meta Theorem Proof:

- $x \in A \cup B \equiv x \in A \lor x \in B$ (Definition of Union)
- $x \in A \cap B \equiv x \in A \land x \in B$ (Definition of Intersection)
- $x \in A^C \equiv \neg(x \in A)$ (Definition of Complement)
- $x \in A / B \equiv x \in A \land \neg(x \in B)$ (Definition of Set Difference)
- $x \in A \oplus B \equiv x \in A \oplus x \in B$ (Definition of Symmetric Difference)

Problem 3

Let A, B, and C be sets. State if the claims below are true or false. If true, prove it. If false, provide a counterexample.

a) $(A \setminus B) \setminus C = A \setminus (B \cap C)$

b) $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$

Problem 3

a) $(A \setminus B) \setminus C = A \setminus (B \cap C)$

Problem 3

a) $(A \setminus B) \setminus C = A \setminus (B \cap C)$

This claim is **false**.

If $A = \{1, 2, 3\}$, $B = \{1, 2\}$, and $C = \{2, 3\}$, then

$A \setminus B = \{1, 2, 3\} \setminus \{1, 2\} = \{3\}$
$(A \setminus B) \setminus C = \{3\} \setminus \{2, 3\} = \varnothing$

$B \cap C = \{1, 2\} \cap \{2, 3\} = \{2\}$
$A \setminus (B \cap C) = \{1, 2, 3\} \setminus \{2\} = \{1, 3\}$

$(A \setminus B) \setminus C = \varnothing \neq \{1, 3\} = A \setminus (B \cap C)$

Problem 3

b) $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$

Problem 3

b) A \ (B ∪ C) = (A \ B) ∩ (A \ C)

This claim is **true**. Here is a proof.

Let x be arbitrary. We can see that the condition of being in the first set is equivalent to being in the second set as follows:

$$
\begin{aligned}
x \in A \backslash (B \cup C) &\equiv x \in A \wedge \neg(x \in (B \cup C)) && \text{[Def of Set Difference]} \\
&\equiv x \in A \wedge \neg(x \in B \vee x \in C) && \text{[Def of Union]} \\
&\equiv x \in A \wedge (x \notin B \wedge x \notin C) && \text{[De Morgan]} \\
&\equiv (x \in A \wedge x \in A) \wedge (x \notin B \wedge x \notin C) && \text{[Idempotency]} \\
&\equiv (x \in A \wedge x \notin B) \wedge (x \in A \wedge x \notin C) && \text{[Associativity/Commutativity]} \\
&\equiv (x \in (A \backslash B)) \wedge (x \in (A \backslash C)) && \text{[Def of Set Difference]} \\
&\equiv x \in (A \backslash B) \cap (A \backslash C) && \text{[Def of Intersection]}
\end{aligned}
$$

Since x was arbitrary, we have shown the sets are the same by the definition of equality.

That's all Folks!