P03_SetTheory

Set Theory

Set Theory — Contents

Set theory: notation

We write S, T, R, … for sets whose elements have some fixed type.

Symbol	**Name**	**Read as**
$x \in S$ / $x \notin S$	membership	"x is (not) in S"
$\{x \mid P(x)\}$	set builder	"the set of x such that $P(x)$"
$S \subseteq T$	subset	"every element of S is in T"
S^c	complement	"the set of x not in S"
$S = T$	set equality	"S and T have the same elements"
$\varnothing$	empty set	"the set with no elements"

Set statements become Lean goals about membership ($x \in S$) and implications.

Set theory: core proof rules

	To prove	To use as hypothesis
$S \subseteq T$	Let $x \in S$ be arbitrary; show $x \in T$.	Apply to a specific $x \in S$ to get $x \in T$.
$S = T$	Show $S \subseteq T$ and $T \subseteq S$ (extensionality).	Replace S by T (or vice versa) anywhere.
S^c	Assume $x \in S$ and derive a contradiction.	From $x \in S^c$ and $x \in S$, conclude anything.

Let's look at this in Lean.

Subsets and Complements: takeaways

1. In Lean, a set `S : Set` α is a predicate $\alpha \to \text{Prop}$. Membership `x` $\in$ `S` is just function application `S x`. So set builder notation, complement membership, and similar definitions hold by `rfl`—they are definitional, not theorems.
2. Subset proofs ($S \subseteq T$) are universally quantified implications. They unfold with `intro`/`apply`—the same pattern as $\forall$ and $\to$ from P02.
3. Set equality uses `ext x` to reduce to pointwise $\leftrightarrow$. The complement chain $x \in S^c \leftrightarrow x \notin S \leftrightarrow \neg(x \in S) \leftrightarrow (x \in S \to \bot)$ holds by `rfl` at every step, so complement reasoning reduces entirely to the negation tactics from P02.

Set Theory — Contents

Intersections and unions

Definitions

$S \cap T = \{x \mid x \in S \ \land \ x \in T\}$ $\qquad$ $S \cup T = \{x \mid x \in S \ \lor \ x \in T\}$

Intersections and unions

Definitions

$S \cap T = \{x \mid x \in S \ \wedge \ x \in T\}$ $\qquad$ $S \cup T = \{x \mid x \in S \ \vee \ x \in T\}$

The proving and using rules are exactly the connective rules from P02, with $x \in S$ replacing P:

	Connective	**Set operation**
$\wedge$ / $\cap$	Prove both; extract either.	$x \in S \cap T$ iff $x \in S$ and $x \in T$.
$\vee$ / $\cup$	Prove one side; case-split to use.	$x \in S \cup T$ iff $x \in S$ or $x \in T$.

Intersections and unions

Definitions

$S \cap T = \{x \mid x \in S \ \wedge \ x \in T\}$ $\qquad$ $S \cup T = \{x \mid x \in S \ \vee \ x \in T\}$

The proving and using rules are exactly the connective rules from P02, with $x \in S$ replacing P:

	Connective	**Set operation**
$\wedge$ / $\cap$	Prove both; extract either.	$x \in S \cap T$ iff $x \in S$ and $x \in T$.
$\vee$ / $\cup$	Prove one side; case-split to use.	$x \in S \cup T$ iff $x \in S$ or $x \in T$.

The algebraic laws of sets—commutativity, associativity, distributivity, De Morgan—are instances of propositional tautologies. Once you unfold definitions with `ext x`, the proof is purely logical.

Let’s look at this in Lean.

Set theory: the reduction principle

Every set-theoretic statement reduces to a logical one. The tactic you need is determined by the logical connective, not the set operation.

Set theory: the reduction principle

Every set-theoretic statement reduces to a logical one. The tactic you need is determined by the logical connective, not the set operation.

Set concept	**Logical core**	**Key tactic**
$x \in \{a \mid P(a)\}$	$P(x)$	`rfl` (definitional)
$S \subseteq T$	$\forall x,\ x \in S \to x \in T$	`intro` / `apply`
$S = T$	$\forall x,\ x \in S \leftrightarrow x \in T$	`ext x`
$S \cap T$	$x \in S\ \wedge\ x \in T$	`constructor` / `obtain`
$S \cup T$	$x \in S\ \vee\ x \in T$	`left`, `right` / `obtain`
S^c	$\neg(x \in S)$	`intro` / `contradiction`

Set Theory — Contents

Set families

We have seen $S \cap T$ and $S \cup T$ for two sets. What about the intersection or union of an arbitrary collection?

Set families

We have seen $S \cap T$ and $S \cup T$ for two sets. What about the intersection or union of an arbitrary collection?

A **set family** $\mathcal{F}$ is a set whose elements are themselves sets. We write:

Notation	**Definition**	**Read as**
$\bigcap \mathcal{F}$	$\{x \mid \forall\, T \in \mathcal{F},\ x \in T\}$	"x is in every set of $\mathcal{F}$"
$\bigcup \mathcal{F}$	$\{x \mid \exists\, T \in \mathcal{F},\ x \in T\}$	"x is in some set of $\mathcal{F}$"

Set families

We have seen $S \cap T$ and $S \cup T$ for two sets. What about the intersection or union of an arbitrary collection?

A **set family** $\mathcal{F}$ is a set whose elements are themselves sets. We write:

Notation	**Definition**	**Read as**
$\bigcap \mathcal{F}$	$\{x \mid \forall\, T \in \mathcal{F},\, x \in T\}$	"x is in every set of $\mathcal{F}$"
$\bigcup \mathcal{F}$	$\{x \mid \exists\, T \in \mathcal{F},\, x \in T\}$	"x is in some set of $\mathcal{F}$"

$\bigcap$ generalises $\cap$ (universal: *for all* members), while $\bigcup$ generalises $\cup$ (existential: *there exists* a member). The same $\forall/\exists$ proof patterns from P02 apply directly.

Note the monotonicity reversal: if $\mathcal{F} \subseteq \mathcal{G}$ (more sets), then $\bigcap \mathcal{G} \subseteq \bigcap \mathcal{F}$ (smaller intersection) but $\bigcup \mathcal{F} \subseteq \bigcup \mathcal{G}$ (larger union).

Set families: proof patterns

1 Membership in family intersections

- Goal shape: $x \in \bigcap \mathcal{F}$.
- Expand to: "for every $T \in \mathcal{F}$, prove $x \in T$".
- Lean pattern: `intro T hT` followed by the required membership proof.

Set families: proof patterns

1 Membership in family intersections

- Goal shape: $x \in \bigcap \mathcal{F}$.
- Expand to: "for every $T \in \mathcal{F}$, prove $x \in T$".
- Lean pattern: `intro T hT` followed by the required membership proof.

2 Membership in family unions

- Goal shape: $x \in \bigcup \mathcal{F}$.
- Expand to: "choose some $T \in \mathcal{F}$ with $x \in T$".
- Lean pattern: `use T`, then prove $T \in \mathcal{F}$ and $x \in T$.

Set families: proof patterns

1 Membership in family intersections

- Goal shape: $x \in \bigcap \mathcal{F}$.
- Expand to: "for every $T \in \mathcal{F}$, prove $x \in T$".
- Lean pattern: `intro T hT` followed by the required membership proof.

2 Membership in family unions

- Goal shape: $x \in \bigcup \mathcal{F}$.
- Expand to: "choose some $T \in \mathcal{F}$ with $x \in T$".
- Lean pattern: `use T`, then prove $T \in \mathcal{F}$ and $x \in T$.

For inclusions between $\bigcap$/$\bigcup$ expressions, first rewrite to membership statements, then solve using the same $\forall$/$\exists$ logic from P02.

Let's look at this in Lean.