How to Write a Formal Mathematical Proof

Prepared by Chuan Zhang

Step 1: Confirm the Task and Formalize the Statement

Before choosing a proof method, restate the problem as a precise mathematical claim.

1.1 Define objects and notation

Clearly define all sets, elements, functions, and parameters. State domains/codomains.

- Example: "Let $f : \mathbb{R} \to \mathbb{R}$ be defined by $f(x) = \ldots$."
- Quantifiers matter: distinguish "for all" ($\forall$) vs. "there exists" ($\exists$).

1.2 Write the target statement in symbols

Turn the English sentence into a formal proposition.

- Example: "For every $n \in \mathbb{N}$, if n is even then n^2 is even," i.e.

$$\forall n \in \mathbb{N} \; (\, n \text{ even} \Rightarrow n^2 \text{ even} \,).$$

1.3 Record given assumptions and goal

Separate hypotheses from what must be shown.

- **Given:** n is even. **Show:** n^2 is even.

Step 2: Choose Proof Tools by Statement Form

$\forall x,\ P(x) \Rightarrow Q(x)$ — Universal Implication

Primary Methods: Direct proof; **contrapositive**; contradiction; case analysis.
When to Use: When forward reasoning from definitions is natural, or when $\neg Q$ is structurally simpler than Q.
Mini-template:

- *Direct:* Assume x with $P(x)$. Show $Q(x)$.
- *Contrapositive:* Assume $\neg Q(x)$. Deduce $\neg P(x)$.

Common Pitfalls: Confusing the converse ($Q \Rightarrow P$) with the contrapositive; incorrect negation of quantified statements.

$\exists x,\ P(x)$ — Existence

Primary Methods: Construction; counting/pigeonhole; extremal arguments.
When to Use: When you can exhibit a witness or force existence via counting principles.
Mini-template:
Take $x = \ldots$; verify that $P(x)$ holds.
Common Pitfalls: Claiming existence without a witness or verification; giving many examples but no general argument.

$\forall n \in \mathbb{N},\ P(n)$ — Inductive Statements

Primary Methods: Induction (weak/strong/structural); minimal counterexample.
When to Use: When the statement depends on size, recurrence, or recursive definitions.
Mini-template:

- Base case: prove $P(n_0)$.
- Inductive step: assume $P(k)$; prove $P(k+1)$.

Common Pitfalls: Missing or incorrect base case; jumping from $P(k)$ to $P(k+2)$ without justification.

$A \iff B$ — Biconditional

Primary Methods: Two separate directions; direct, contrapositive, or contradiction.
When to Use: When definitions translate between forms and each direction has a natural route.
Mini-template:

- $(\Rightarrow)$: Assume A, show B.
- $(\Leftarrow)$: Assume B, show A.

Common Pitfalls: Proving only one direction; mixing both directions in a single chain of logic.

$A \subseteq B$ or $A = B$ — Set Relations

Primary Methods: Element chasing; two-inclusion method; **contrapositive** for subset.
When to Use: When it is clearer to reason about individual elements.
Mini-template:

- Let $x \in A$. By $\ldots$, show $x \in B$.
- For equality: prove both $A \subseteq B$ and $B \subseteq A$.

Common Pitfalls: Thinking "large overlap" implies subset; forgetting to state "for any x".

$\exists! x,\ P(x)$ — Existence and Uniqueness

Primary Methods: Split into existence and uniqueness.
When to Use: When you can construct a witness and show any two must coincide.
Mini-template:

- Existence: give x with $P(x)$.
- Uniqueness: suppose $P(x)$ and $P(y)$, then prove $x = y$.

Common Pitfalls: Proving existence but not uniqueness; confusing "at most one" with "exactly one".

Impossibility / Non-existence

Primary Methods: Contradiction; invariants; parity/coloring/modular arguments.
When to Use: When process constraints or parity/coloring obstructions apply.
Mini-template:
Suppose, for contradiction, an object exists. Track an invariant $\mathcal{I}$ and reach a contradiction.
Common Pitfalls: Using examples as "evidence" only; failing to state or prove the invariant.

Algorithm Correctness / Complexity

Primary Methods: Loop invariants (init/maintain/terminate); exchange argument; induction; potential method.
When to Use: For loops, recurrences, greedy choices, or amortized bounds.
Mini-template:
Invariant I: holds before the loop; one iteration preserves I; upon termination, I implies the post-condition.
Common Pitfalls: Saying "it obviously works"; vague invariant; termination not linked to the goal.

Step 3: Structure and Language (with Positive and Negative Examples)

Each aspect below includes a correct pattern and multiple common pitfalls.

3.1 Frame the proof clearly

Aspect	Good Example	Bad Examples (and why)
State the claim and plan	*Proposition.* $\forall n \in \mathbb{N}$, if n is even then n^2 is even. *Proof.* Let n be even; then $n = 2k$ for some $k \in \mathbb{N}$. We compute $n^2 = (2k)^2 = 4k^2 = 2(2k^2)$, hence even. $\square$	(1) "$n = 2 \Rightarrow n^2 = 4$; $n = 4 \Rightarrow 16$; so always even." *(only examples)* (2) "This is true because squares are even." *(assertion without reason)* (3) "$n^2 = 4k^2$, so $n = 2k$." *(backward causation / converse)*

3.2 Use precise logical language

Aspect	Good Example	Bad Examples (and why)
Connectives and tone	"Since $x \in A$ and $A \subseteq B$, it follows that $x \in B$."	(1) "x is in A, so it should be in B." *(hedging)* (2) "Obviously x is in B!" *(rhetoric instead of reason)* (3) "$x \in A$, $A \subseteq B$, $x \in B$." *(no connective; reader must infer)*

3.3 Justify every inference

Aspect	Good Example	Bad Examples (and why)
No leaps; cite reasons	*Induction:* Assume $P(k)$. For $n = k + 1$, by the recurrence ... hence $P(k + 1)$.	(1) "$n = k + 1$ works similarly." *(missing argument)* (2) "n^2 is even, so n is even." *(using conclusion as premise)* (3) "Even squares are even." *(circular reasoning)*

3.4 Keep notation consistent and define everything

Aspect	Good Example	Bad Examples (and why)
Consistency and definitions	"Let $f : \mathbb{R} \to \mathbb{R}$. For any $x \in \mathbb{R}$, we have ..."	(1) Switch f to g later but mean the same function. *(symbol clash)* (2) "Let $x \in S$, then $x \in T$." with S, T undefined. *(undefined terms)* (3) Jammed formulas with no words or spacing. *(unreadable)*

Step 4: Self-Check Before You Submit

Use this quick checklist after drafting your proof.

4.1 Logical completeness

- Does each line follow from previous lines (or stated facts) with a named reason?
- Can you cover the next line and still derive it from what remains above?

4.2 Assumptions vs. conclusions

- Did you explicitly use all the given hypotheses?
- Did you introduce any new symbol without definition?

4.3 Edge cases and scope

- Have you addressed base cases, boundary values, empty sets, smallest graphs, etc.?
- Did you accidentally prove only special cases or examples?

4.4 Form and readability

- Are paragraphs structured, connectives used (*thus, hence, therefore*)?
- Is the notation consistent? Are statements quantifier-correct?