Fundamentals of Mathematics Lecture 2: Proof Techniques

Guan-Shieng Huang

National Chi Nan University, Taiwan

Spring, 2008

Reasoning Methods

To derive knowledge from assumptions, other facts, or previous results through

- Deduction;
- Induction.

We focus on deduction techniques in this lecture.

Reasoning Methods

To derive knowledge from assumptions, other facts, or previous results through

- Deduction;
- Induction.

We focus on deduction techniques in this lecture.

Note

Abduction is not a rigid reasoning method.

Deduction

To infer specific cases from general cases.

Deduction Techniques

- Proof Patterns
- Inference Rules
- The Negation of a Proposition

Proof Patterns I

- Proofs for $p \to q$
  1. direct proof: Assume p, then derive q from p and other facts.
     - proof by cases: Assume p, and we know $p \equiv p_1 \vee p_2 \vee \cdots \vee p_k$. Then establish $p_i \to q$ for $1 \le i \le k$. Hence we get $p \to q$.
  2. indirect proof (proof by contraposition): Assume $\neg q$, and establish $\neg q \to \neg p$. Then conclude $p \to q$.
  3. proof by contradiction: Assume p and $\neg q$, and then get a contradiction. Hence $p \to q$.

Proof Patterns II

Example

For all integers m and n, if m and n are even, then $m + n$ is even. (direct proof)

Example

For all odd integers n, the number $n^2 - 1$ is divisible by 8.

Example

For all integers n, if n^2 is even , then n is even. (contraposition)

Example

$\sqrt{2}$ is irrational. (contradiction)

Proof Patterns III

Fundamentals of Mathematics Lecture 2: Proof Techniques

Guan-Shieng Huang

Deduction Techniques

Induction Techniques

Abduction Techniques

Reduction Techniques

References

- $p \to (q \lor r)$: Prove $p \land \neg q \to r$, or $p \land \neg r \to q$.

Example

For all integers a and p, if p is prime, then either p is a divisor of a, or a and p have no common factor greater than 1.

Example

For all integers n, $n^2 - 1$ is either divisible by 8 or relative prime to 8.

Proof Patterns IV

- $p_1, p_2, \ldots, p_k$ are equivalent: (Proof by cycle implications) Prove $p_1 \to p_2$, $p_2 \to p_3$, …, $p_k \to p_1$.

Example

For all integers n, the following statements are equivalent:

1. n is even;
2. n^2 is even;
3. n^k is even for all integers $k \geq 1$.

$(1 \to 3 \to 2 \to 1)$

Proof Patterns V

- $(\forall_{x \in D}) P(x)$:
  1. direct proof: Let x be an arbitrary element in D. Then derive $P(x)$ is true.
  2. proof by contradiction: Assume there is some $c \in D$ such that $P(c)$ is false. Show that a contradiction results.

Example

For all integers n, if n is even, then n^2 is even.

Example

Every finite acyclic graph must have a source.

Proof Patterns VI

- $(\exists_{x \in D}) P(x)$:
    1. constructive proof: Try to find a c such that $P(c)$ is true.
    2. nonconstructive proof: Derive the existence of x by mathematical facts (e.g., counting or the pigeon-hole principle).
    3. proof by contradiction: Assume there is no $x \in D$ such that $P(x)$ is true, and derive a contradiction.

Example

There exists a number that is not rational. $(\sqrt{2})$

Example

Given any seven integers $a_1, a_2, \ldots, a_7$, there always exist $1 \le i < j \le 7$ such that $a_i + a_{i+1} + \cdots + a_j$ is a multiple of 7.

Proof Patterns VII

- $(\forall_{x \in D_1})(\exists_{y \in D_2})P(x, y)$:
  1. constructive proof: Let x be an arbitrary element of D. Construct $y \in D$ as a function of x, and show that $P(x, y)$ is true.
  2. nonconstructive proof

Example

Given any integer n, there is an integer m with $m > n$.

Example

Given a natural number n, there is always a prime number p that is greater than n.

Proof Patterns VIII

Example

Every finite acyclic graph must have a source.

Inference Rule

- Inference rules are used to derive new facts from previous results, assumptions, or other facts.
- What is a sound inference step? It should hold for all models, with no exception.

Inference Rule

- Inference rules are used to derive new facts from previous results, assumptions, or other facts.
- What is a sound inference step? It should hold for all models, with no exception.

Example

$p \to q \vdash \neg p \to \neg q$ is not a sound inference, since it can be $M \models \neg p \wedge q$. E.g., p: x is an even number, q: x is a number.

Some Common Inference Rules

- modus ponens (method of affirming): $p \rightarrow q, p \vdash q$
- modus tollens (method of denying): $p \rightarrow q, \neg q \vdash \neg p$
- hypothetical syllogism: $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$
- dilemma by cases: $p \vee q, p \rightarrow r, q \rightarrow r \vdash r$
- conditional proof: $p, p \wedge q \rightarrow r \vdash q \rightarrow r$
- rules of contradiction: $\neg p \rightarrow \perp \ \vdash p$
- instantiation-$\forall$: $(\forall_{x \in D} Q(x) \vdash Q(a)$ where $a \in D$
- generalization-$\forall$: $Q(a) \vdash (\forall_{x \in D})Q(x)$ where a is an arbitrary chosen element in D
- speciation-$\exists$: $(\exists_{x \in D})Q(x) \vdash Q(a)$ for some $a \in D$
- generalization-$\exists$: $Q(a) \vdash (\exists_{x \in D})Q(x)$ for some $a \in D$

Contraposition: Revisited

- Assume $\neg q$, and establish $\neg q \to \neg p$. Then conclude $p \to q$.
- modus tollens (method of denying): $p \to q, \neg q \vdash \neg p$
- $p \to q \equiv \neg q \to \neg p$

Remark

A common error is to conclude $\neg p \to \neg q$ *from* $p \to q$.

Contraposition: Revisited

- Assume $\neg q$, and establish $\neg q \rightarrow \neg p$. Then conclude $p \rightarrow q$.
- modus tollens (method of denying): $p \rightarrow q, \neg q \vdash \neg p$
- $p \rightarrow q \equiv \neg q \rightarrow \neg p$

Remark

A common error is to conclude $\neg p \rightarrow \neg q$ *from* $p \rightarrow q$.

Example

如果考試作弊，學期成績一定不及格。

錯誤的結論是：因爲沒做弊，所以成績一定及格。

The Negation of a Proposition

- $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- $\neg(p \wedge q) \equiv \neg p \vee \neg q$
- $\neg(p \rightarrow q) \equiv p \wedge \neg q$
- $\neg\forall_x P(x) \equiv \exists_x \neg P(x)$
- $\neg\exists_x P(x) \equiv \forall_x \neg P(x)$

The Negation of a Proposition

- $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- $\neg(p \wedge q) \equiv \neg p \vee \neg q$
- $\neg(p \rightarrow q) \equiv p \wedge \neg q$
- $\neg\forall_x P(x) \equiv \exists_x \neg P(x)$
- $\neg\exists_x P(x) \equiv \forall_x \neg P(x)$

Example

A function $f(x)$ is continuous at $x = a$ iff for every $\epsilon > 0$ there exists $\delta > 0$ such that $|f(x) - f(a)| < \epsilon$ for all $|x - a| < \delta$.

The Negation of a Proposition

- $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- $\neg(p \wedge q) \equiv \neg p \vee \neg q$
- $\neg(p \rightarrow q) \equiv p \wedge \neg q$
- $\neg\forall_x P(x) \equiv \exists_x \neg P(x)$
- $\neg\exists_x P(x) \equiv \forall_x \neg P(x)$

Example

A function $f(x)$ is continuous at $x = a$ iff for every $\epsilon > 0$ there exists $\delta > 0$ such that $|f(x) - f(a)| < \epsilon$ for all $|x - a| < \delta$.

$C(f, a) :=$
$(\forall_{\epsilon>0})(\exists_{\delta>0})(\forall_{x\in\mathbb{R}})\{|x-a| < \delta \rightarrow |f(x) - f(a)| < \epsilon\}$
$\neg C(f, a) :=$
$(\exists_{\epsilon>0})(\forall_{\delta>0})(\exists_{x\in\mathbb{R}})\{|x-a| < \delta \wedge |f(x) - f(a)| \not< \epsilon\}$

Induction

- The process to discover general rules or principles from particular facts and examples

We will discuss induction methods in latter lectures.

Abduction

To infer the cause from the observation.

$$p \rightarrow q, q \vdash p$$

Since the cause is not unique, the conclusion may be incorrect.

- Statistical inferences
- Belief

It is a common reasoning method in our daily life.

Reduction

To transform solving a problem into solving another problem.

- Karp reduction (or many-one reduction)
- Turing reduction (or Cook reduction)
- Self-reduction (especially useful when combined with induction)

Karp Reduction

$A \leq_M B$

- Find a mapping that can translate any instances of A to instances of B such that A can be solved once B can be solved.
- Let f be the mapping. We need to establish
  1. For any $x \in A$, $f(x) \in B$.
  2. $f(x)$ can be solved.
  3. The solution of $f(x)$ can be used to construct the solution of x.

Karp Reduction

$A \leq_M B$

- Find a mapping that can translate any instances of A to instances of B such that A can be solved once B can be solved.
- Let f be the mapping. We need to establish
  1. For any $x \in A$, $f(x) \in B$.
  2. $f(x)$ can be solved.
  3. The solution of $f(x)$ can be used to construct the solution of x.

Turing Reduction

$A \leq_T B$

- Take B as an oracle (or a black box). Any question of B can be answered.
- Solve $x \in A$ by asking finite questions of B.
  1. Given any $x \in A$, construct $q_1(x), q_x(x), \ldots, q_k(x) \in B$.
  2. Solve $q_i(x)$, and get the corresponding answer $a_i(x)$.
  3. Solve x based on $(q_1(x), a_1(x)), \ldots, (q_k(x), a_k(x))$.

Self-Reduction

$A \le A$

- Reduce a problem into itself. The problem size becomes smaller.
  1. $A \le_T A$: (divide-and-conquer)
     - Break $x \in A$ into several subproblems $x_1, \ldots, x_k$.
     - Recursively solve each subproblem x_i.
     - Merge the results of $x_1, \ldots, x_k$, and get a solution of x.
  2. $A \le_M A$: (prune-and-search)
     - Reduce $x \in A$ to $x' \in A$ such that the size of x' is smaller than x.
     - Solve x' by applying the same reduction technique.
     - Obtain the result of x from the result of x'.
- Especially useful when combined with induction. We will revisit it when discussing induction.

Summary

- Reasoning is based on deduction and induction.
- Deduction reasons from general to special and induction reasons from special to general.
- Abduction infers the cause, but the conclusion may be incorrect. Hence we never use abduction in mathematical proofs.
- An inference rule (or step) should hold for all cases.
- Reduction transforms a problem into another; once the latter is solved, so does the original one.

國立暨南國際大學

References I

K. H. Rosen (editor), Handbook of Discrete and Combinatorial Mathematics, CRC Press LLC, 2000.

L. Lovász, J. Pelikán, K. Vesztergombi, Discrete Mathematics: Elementary and Beyond, Springer-Verlag, 2003.

D. C. Kozen, Theory of Computation, Springer-Verlag, 2006.