

# Lecture 2

Alex Hassett  
Applied Analysis

May 30, 2025

**Definition 2.1.** A proof is a formal argument regarding a statement. The aim of a proof is to show that an argument is either true or false using the formal rules of mathematics. Expressed symbolically, to directly prove the statement  $p \implies q$  true, one supposes that  $p$  is true,  $p \equiv T$ , and then shows that  $q$  is also true. The reason that one must show that  $q$  is true after supposing that  $p$  is true, is because now the statement  $p \implies q$  has become “if true, then  $q$ ”. For this statement to be true  $q$  must be true.

**Theorem 2.2.** Let  $p$  and  $q$  be statements. Then  $p \implies q$  is also a statement. Suppose that  $p \equiv T$ , then  $p \implies q \equiv T$  only if  $q \equiv T$ .

*Proof.* Suppose that  $p \equiv T$ . Then by Theorem 1.1, we have

$$\begin{aligned} p \implies q &\equiv \neg p \vee q \\ &\equiv \neg T \vee q \\ &\equiv F \vee q \\ p \implies q &\equiv q \end{aligned}$$

Therefore, if  $p \equiv T$ , then  $p \implies q \equiv q$ . □

**Theorem 2.3.** To prove the statement  $p \implies q$  true via contradiction, one supposes the negation,  $p \wedge \neg q$ , to be true, then arrive at a contradiction ( $F$ ). The most common contradiction is  $q \wedge \neg q \equiv F$ .

*Proof.* Let  $p \implies q$  be a statement. Then its negation is  $p \wedge \neg q$ . Suppose the negation is true, then  $p \equiv T$  and  $q \equiv F$ . Now, suppose this leads to some contradiction. Then  $p \wedge \neg q \implies F$ . By Theorem 1.1, we have

$$\begin{aligned} ((p \wedge \neg q) \implies F) \implies (p \implies q) &\equiv (\neg(p \wedge \neg q) \vee F) \implies (\neg p \vee q) \\ &\equiv (\neg p \vee \neg(\neg q) \vee F) \implies (\neg p \vee q) \\ &\equiv (\neg p \vee q \vee F) \implies (\neg p \vee q) \\ &\equiv (\neg p \vee q) \implies (\neg p \vee q) \\ &\equiv \neg(\neg p \vee q) \vee (\neg p \vee q) \\ &\equiv \neg(\neg p) \wedge \neg q \vee (\neg p \vee q) \\ &\equiv p \wedge \neg q \vee (\neg p \vee q) \\ &\equiv p \wedge \neg q \vee (q \vee \neg p) \\ &\equiv p \wedge (\neg q \vee q) \vee \neg p \\ &\equiv p \wedge T \vee \neg p \\ &\equiv p \vee \neg p \\ &\equiv T \end{aligned}$$

Therefore, we have  $p \implies q \equiv T$ . □

**Theorem 2.4.** To prove the statement  $p \implies q$  true indirectly, one proves the statement  $r \implies (p \implies q)$  true, then proves that  $r$  is true, i.e. that  $r \equiv T$ .

*Proof.* Suppose  $r \implies (p \implies q) \equiv T$  and suppose  $r \equiv T$ . Then by Theorem 1.1, we have

$$\begin{aligned} r &\implies (p \implies q) \equiv T \\ \neg r \vee (p \implies q) &\equiv T \\ \neg T \vee (p \implies q) &\equiv T \\ F \vee (p \implies q) &\equiv T \\ p \implies q &\equiv T, \end{aligned}$$

as desired.  $\square$

**Definition 2.5.** There are multiple proven methods to prove  $p \implies q$  true. We say two methods of proof are different if they have different suppositions, but equivalent conclusions, i.e. they have different suppositions, but both methods conclude  $p \implies q \equiv T$ .

**Definition 2.6.** A logical system (also called a formal system) is a formal framework used to carry out logical reasoning. It consists of the following components:

1. A set of symbols (an Alphabet) used to construct formulas. These include
  - Variables:  $x, y, z, \dots$
  - Logical connectives:  $\neg, \wedge, \vee, \implies, \iff$
  - Quantifiers:  $\forall, \exists$
  - Parenthesis:  $(, )$
2. Formation rules (Grammar): Rules that determine how to combine symbols to create logical expressions. These ensure syntactic validity.
3. Interpretation rules (semantics): Defines what formulas mean, assigning truth values to formulas based on interpretations of their symbols.
4. Inference rules: A set of logical rules that specify how to derive new formulas (conclusions) from existing ones (premises).
5. Axioms: A set of foundational statements taken to be true without proof within the system. They serve as the starting point for logical deductions.
6. Proof system (or deductive system): A method of deriving conclusions using axioms and inference rules. If a formula can be derived from the axioms using the inference rules, it is said to be provable in the system.

**Definition 2.7.** The domain of discourse, or just domain, is the set of all objects under consideration in a logical system. For example, if discussing real numbers, the domain might be  $\mathbb{R}$ .

**Definition 2.8.** A logical formula is a syntactically correct expression built from variables, logical connectives ( $\neg, \wedge, \vee, \implies$ ), quantifiers ( $\forall, \exists$ ), and possibly predicates or relations.

**Definition 2.9.** A variable is a symbol that stands for an unspecified element of a given set. In first-order logic a variable can be either bound or free. A free variable is one that is not bound by any quantifier (like  $\forall$  or  $\exists$ ) in a logical expression (statement). A bound variable is a variable that is within the scope of a quantifier, such as  $\forall$  or  $\exists$ . A bound variable does not refer to a specific value, but rather represents all, or some, values in the domain that it is bound to.

**Definition 2.10.** A predicate is a function or relation that returns true or false for elements from the domain. It expresses a property or relation. For example,  $P(x) \mid x > 2$ .

**Definition 2.11.** A quantifier is a symbol used to indicate the scope of a variable over the domain of discourse.

**Definition 2.12.** The scope of a variable refers to the part of the logical formula in which that variable is bound by a quantifier and can be referenced meaningfully as such. The formal definition of the scope of a quantifier (like  $\forall x$  or  $\exists x$ ) is the subformula over which that quantifier has authority, i.e. where the variable  $x$  is bound and interpreted as ranging over the domain of discourse.

**Remark 2.13.** The idea expressed in Definition 2.12 has many parallels in programming. The scope of a variable can be thought of like declaring a variable inside a function in programming. Just like a variable declared inside a function can't be accessed outside of it, a bound variable only "makes sense" within the scope of the quantifier that binds it. As an example, consider the formula:

$$\forall x(P(x) \implies Q(x)) \wedge R(x).$$

The scope of  $\forall x$  is only the part (subformula):

$$(P(x) \implies Q(x)).$$

So within that part,  $x$  is bound. But in  $R(x)$ , the  $x$  is not bound by any quantifier, i.e. it's a free variable there. Also note that the  $x$  in  $(P(x) \implies Q(x))$  and the  $x$  in  $R(x)$  both have the same domain.

**Definition 2.14.** An open formula is a logical formula with free variables (i.e., not all variables are bound by quantifiers). An open formula does not yet have a truth value. For example,  $x > 2$  is open because  $x$  is free.

**Definition 2.15.** A closed formula is a logical formula where all variables are bound by quantifiers. It does have a definite truth value. For example,  $\forall x \in \mathbb{R}, x^2 \geq 0$ .

**Definition 2.16.** A statement (proposition) is a closed formula that is either true or false. Less formally, a proposition (statement) is a declarative sentence that expresses a truth value.

**Definition 2.17.** Logical operators (connectives) are symbols used to combine logical formulas or propositions. The standard ones are

- Negation:  $\neg$
- Conjunction:  $\wedge$
- Disjunction:  $\vee$
- Implication:  $\implies$
- Biconditional:  $\iff$

**Definition 2.18.** A formula or statement is self-contained if it does not depend on any undefined variables or outside assumptions. This means every variable is bound, every symbol is defined, and it can be evaluated or reasoned about in isolation. Complete proposition is another phrase for self-contained statement.

**Definition 2.19.** The statement "If  $P$ , then  $Q$ " means that  $P$  is a sufficient condition for  $Q$ , i.e. if  $P$  is true, then  $Q$  must also be true. Note that  $Q$  can still be true without  $P$  being true. This is formally expressed as  $P \implies Q$ .

**Definition 2.20.** The statement " $Q$  only if  $P$ " means that  $P$  is a necessary condition for  $Q$ , i.e. if  $Q$  is true, then  $P$  must also be true. Note that  $P$  being true doesn't necessarily make  $Q$  true. This is formally expressed as  $Q \implies P$ .

**Definition 2.21.** The statement " $P$  if and only if  $Q$ " means that  $P$  is both necessary and sufficient for  $Q$ . This is formally expressed as  $P \iff Q$  (read " $P$  if and only if  $Q$ "), i.e.  $P \implies Q$  and  $Q \implies P$ .