

2610 Discrete Mathematics for Computer Science

Department of Computer Science
University of Georgia
Athens, GA 30602

Instructor: Liming Cai

<http://www.cs.uga.edu/~cai/courses/discmath/>

Organization of lectures:

Part 1 (*Chapters 1-4*): Introduction to logic and proofs.

Part II (*Chapters 5-7*): Advanced counting

Part III (*Chapters 8-10*): Relation, graph, and tree

Part IV (*Chapters 11-12*): Boolean algebra and computational models

Part I. Introduction to logic and proofs

Chapter 1. The foundations: logic and proofs

Chapter 2. Basic structures: sets, functions, sequences, and sums

Chapter 3. The fundamentals: algorithms, the integers, and matrices

Chapter 4. Induction and recursion

Why is logic so important for CS?

- logic is directly used in programming languages and circuits
- logic reasoning is needed for algorithm design, program analysis, system design, and other creative work in CS

to make decisions as what is correct, useful, feasible, reliable, etc

Chapter 1. The foundations: logic and proofs

1.1 propositional logic

proposition: a declarative sentence that is of either *true* or *false* value

example:

DC is the capital of the USA.

Read it carefully.

Today is not cold.

We use letters p, q, r, \dots to denote propositions and

Truth value of a proposition is denoted T (resp. F) if it is a true (resp. false) proposition.

propositional logic: a calculus or mathematical system to deal with propositions.

As the number system, it needs *operators* to produce new propositions (via compounding).

logic operators: *negation, conjunction, disjunction, implication, and bi-conditional*.

examples: $\neg p$, $p \wedge q$, $p \vee q$, $p \rightarrow q$, $p \leftrightarrow q$

truth table for a proposition: displays a relationship between truth values of the proposition and truth values of the involved propositions.

truth tables for $\neg p$, $p \wedge q$, $p \vee q$, $p \rightarrow q$, $p \leftrightarrow q$

other examples

1.2 Propositional equivalences

equivalent propositions: if they are evaluated to the same truth value.

precedence of logical operators: \neg , \wedge , \vee , \rightarrow , \leftrightarrow

We can use truth tables to prove equivalent propositions.

example: $p \rightarrow q$ and $\neg p \vee q$ are equivalent.

Table 6 page 24, laws for propositional equivalences, e.g.,

De Morgan's laws: $\neg(x \wedge y) \equiv \neg x \vee \neg y$

distributive laws: $(x \wedge y) \vee z \equiv (x \vee z) \wedge (y \vee z)$

associative laws: $(x \vee y) \vee z \equiv x \vee (y \vee z)$

We can also use table 6 on page 24 and other proved equivalences to prove two propositions are equivalent.

Table 7, page 25, logical equivalences involving implications

Table 8, page 25, logical equivalences involving bi-conditionals.

tautology: a compound proposition that is always true

contradiction: a compound proposition that is always false

p and q are *logically equivalent* when proposition $p \leftrightarrow q$ is a tautology. Denote this $p \equiv q$.

1.3 Predicates and quantifiers

predicate: a property that the subject of a statement can have.

example: *Friday is not cold*

"*x is not cold*" = $P(x)$ where

P stands for "is not cold".

x is a variable.

more example: $Q(x, y) = "x \text{ is } y^2"$.

in general, $P(x_1, \dots, x_n)$ is a propositional function

quantification: to create a proposition from a propositional function.

universal quantification: allows to assert that a predicate is true for all domain values of the involved variable

$\forall xP(x)$ – what if $P(x)$ stands for “ $x^2 \geq x$ ”?

existential quantification: allows to assert that a predicate is true for some domain value of the variable.

$\exists xP(x)$

bound variable

free variable

scope of a quantifier

negation of a quantifier

1.4 Nested quantifiers

examples: p50-57

rules about universal quantifiers and existential quantifiers

$$\forall x(A(x) \wedge B(x)) \equiv \forall xA(x) \wedge \forall xB(x)$$

$$\exists x(A(x) \vee B(x)) \equiv \exists xA(x) \vee \exists xB(x)$$

$$\neg\forall xA(x) \equiv \exists\neg A(x)$$

$$\neg\exists xA(x) \equiv \forall\neg A(x)$$

1.5 Rules of inferences

A *proof* consists of a sequence of true statements, each a fact or derived from previous statements.

Statements are also called *axioms* or *postulates*

Rules of inference are used to derive axioms.

“ \rightarrow ” will be very useful logical operator for making up rules.

Rules of inference for propositional logic:

$$\begin{array}{l} p \\ \underline{p \rightarrow q} \\ \Rightarrow q \end{array} \quad \text{tautology } (p \wedge (p \rightarrow q)) \rightarrow q$$

Table 1 page 66

(rules of inference for propositional logic)

more examples

$$\begin{array}{l} \neg q \\ \underline{p \rightarrow q} \\ \Rightarrow \neg p \end{array} \quad \text{tautology } (\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$$

$$\begin{array}{l} p \rightarrow q \\ \underline{q \rightarrow r} \\ \Rightarrow p \rightarrow r \end{array} \quad \text{tautology } [(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$$

$$\begin{array}{l} p \vee q \\ \underline{\neg p} \\ \Rightarrow q \end{array} \quad \text{tautology } ((p \vee q) \wedge \neg p) \rightarrow q$$

$$\begin{array}{l} p \vee q \\ \underline{\neg p \vee r} \\ \Rightarrow q \vee r \end{array} \quad \text{tautology } ((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$$

Using rules of inference to build arguments

examples: pages 67-68

Common fallacies:

$[(p \rightarrow q) \wedge \neg p] \rightarrow \neg q$ is not a tautology.

$[(p \rightarrow q) \wedge q] \rightarrow p$ is not a tautology.

Rules of inference for quantified statements

table 2, p70.

universal instantiation

universal generalization

existential instantiation

existential generalization

examples

1.6 Introduction to Proofs

methods for proving theorems

A theorem may be stated as

$$p \rightarrow q$$

or

$$\forall x \exists y (P(x, y) \rightarrow Q(x, y))$$

(1) direct proof

To directly establish the implication $p \rightarrow q$ by showing if p is true, then q is true.

Note that we do not need to show the cases when p is false!

All we need to do is to show:

if p is true, q has to be true.

(2) indirect proof

To prove $p \rightarrow q$,

- first assume $\neg q$
- infer $\neg p$
- draw conclusion $\neg q \rightarrow \neg p$ by

by using *contraposition* $p \rightarrow q \equiv \neg q \rightarrow \neg p$

also called *proof by contradiction*

(3) vacuous proof (when $p = F$)

example: $P(n)$ is " $\forall n$ if $n > 1$ then $n^2 > n$ ",
show it is true for $n = 0$.

(4) proof by contradiction

To show a statement p to be true,

- first assume $\neg p$,
- get $(r \wedge \neg r)$, a contradiction
- conclude that p has to be true.

(5) Counter examples

useful in disproving a statement.

(6) Mistakes in proofs

- logical errors,
- circular reasoning

1.7 Proof Methods and Strategy

(1) proof strategy; try direct proof first, and try indirect proof.

examples

(2) proof by cases

$$(p_1 \vee p_2 \vee p_3) \rightarrow q \equiv (p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge (p_3 \rightarrow q)$$

(3) proof of equivalence

$$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

(4) existence proofs $\exists x P(x)$

constructive proof

non-constructive proof

(5) uniqueness proofs

$$\exists x P(x) \wedge \forall y (y \neq x \rightarrow \neg P(y))$$

(6) counter examples