Discrete Structures

Joy Chandra Mukherjee

August 8, 2023

Chapter 1

Propositional Logic or Propositional Calculus

The rules of logic give precise meaning to mathematical statements. If we prove a mathematical statement is true, then we call it a theorem. A proof makes up a correct mathematical argument for a theorem. In this chapter, we will explain what makes up a correct mathematical argument and introduce tools to construct these arguments. We will develop an arsenal of different proof methods that will enable us to prove many different types of results. After introducing many different methods of proof, we will introduce several strategies for constructing proofs.

Logic is the basis of all mathematical reasoning, and of all automated reasoning. It has various practical applications in computer science.

- To the design of computing machines or computer circuits
- To the specification of systems
- To verify the correctness of a program, i.e., the computer program produces the correct output for all possible input value.
- To establish the security of a system
- To artificial intelligence

- To construct a computer program
- To programming languages, and to other areas of computer science
- Many problems in robotics, software testing, artificial intelligence planning, computer-aided design, machine vision, integrated circuit design, scheduling, computer networking, puzzles, and genetics, can be modeled in terms of propositional satisfiability.

Definition 1 (Proposition) *A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both. The variables that represent propositions, are called* **propositional variables or sentential variables**. *The conventional letters used for propositional variables are* $p, q, r, s, \ldots$.

The truth value of a proposition is true, denoted by T, if it is a true proposition, and the truth value of a proposition is false, denoted by F, if it is a false proposition. Propositions that cannot be expressed in terms of simpler propositions are called atomic propositions.

The area of logic that deals with propositions is called the **propositional calculus** *or* **propositional logic**. *It was first developed systematically by the Greek philosopher Aristotle more than 2300 years ago.*

New propositions, called compound propositions, are formed from one or more existing atomic or compound propositions by combining them using logical operators. These methods were discussed by the English mathematician George Boole in 1854 in his book **The Laws of Thought**.

Definition 2 (Negation) *Let p be a proposition. The negation of p, denoted by $\sim p$, $\neg p$ (also denoted by $\overline{p}$), is the statement "It is not the case that p." The proposition $\sim p$ or $\neg p$ is read "not p". The truth value of the negation of p, $\sim p$ or $\neg p$, is the opposite of the truth value of p.*

Definition 3 (Conjunction) *Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition "p and q". The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.*

p	$\neg p$
F	T
T	F

Table 1.1: Truth Table of Unary Logical Operator

Definition 4 (Disjunction/Inclusive Or) *Let p and q be propositions. The disjunction of p and q, denoted by $p \vee q$, is the proposition "p or q". The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.*

Definition 5 (Exclusive Or) *Let p and q be propositions. The exclusive or of p and q, denoted by $p \oplus q$ (or p XOR q), is the proposition that is true when exactly one of p and q is true and is false otherwise.*

Definition 6 (Implication) *Let p and q be propositions. The conditional statement $p \to q$ is the proposition "if p, then q". The conditional statement $p \to q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \to q$, p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence). A variety of terminology is used to express the conditional statement $p \to q$.*

- *"if p, then q"*
- *"p implies q"*
- *"if p, q"*
- *"p only if q"*
- *"p is sufficient for q"*
- *"a sufficient condition for q is p"*
- *"q if p"*
- *"q whenever p"*
- *"q when p"*
- *"q is necessary for p"*

- *"a necessary condition for p is q"*
- *"q follows from p"*
- *"q unless $\neg p$"*
- *"q provided that p"*

Definition 7 (Converse, Contrapositive, and Inverse) *We can form some new conditional statements starting with a conditional statement $p \to q$. In particular, there are three related conditional statements that occur so often that they have special names.*

- *The proposition $q \to p$ is called the converse of $p \to q$.*
- *The proposition $\neg q \to \neg p$ is called the contrapositive of $p \to q$.*
- *The proposition $\neg p \to \neg q$ is called the inverse of $p \to q$.*

Definition 8 (Bi-Implication) *Let p and q be propositions. The biconditional statement $p \longleftrightarrow q$ is the proposition "p if and only if q". The biconditional statement $p \longleftrightarrow q$ is true when p and q have the same truth values, and is false otherwise.*

p	q	p $\wedge$ q	p $\vee$ q	p $\oplus$ q	p $\to$ q	p $\longleftrightarrow$ q
F	F	F	F	F	T	T
F	T	F	T	T	T	F
T	F	F	T	T	F	F
T	T	T	T	F	T	T

Table 1.2: Truth Table of Binary Logical Operators

Precedence of Logical Operators: $() \succ \neg \succ \wedge \succ \vee \succ \to \succ \longleftrightarrow$

Definition 9 (Bitwise Operations) *Computers represent information using bits. The word bit comes from binary digit, because zeros and ones are the digits used in binary representations of numbers. The well-known statistician*

John Wilder Tukey introduced this terminology in 1946. We will use a 1 bit to represent the truth value T (true) and a 0 bit to represent the truth value F (false). A variable is called a Boolean variable if its value is either true or false. Consequently, a Boolean variable can be represented using a bit.

A bit string is a sequence of zero or more bits. The length of this string is the number of bits in the string. Information is often represented using bit strings, which are lists of zeros and ones. When this is done, bitwise operations on the bit strings can be used to manipulate the information.

x	y	x ∧ q	p ∨ q	p ⊕ q
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Table 1.3: Truth Table of Bitwise Binary Operators

Definition 10 (Tautology/Valid) *A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a tautology.*

Definition 11 (Contradiction/Unsatisfiable) *A compound proposition that is always false, no matter what the truth values of the propositional variables that occur in it, is called a contradiction.*

Definition 12 (Contingency) *A compound proposition that is neither a tautology nor a contradiction is called a contingency.*

Definition 13 (Satisfiable) *A compound proposition is satisfiable if there is an assignment of truth values to its variables that makes it true (that is, when it is a tautology or a contingency).*

1.1 Propositional Equivalences

Definition 14 (Logical Equivalence) *The compound propositions p and q are called logically equivalent, denoted as $p \equiv q$ or $p \iff q$, if $p \longleftrightarrow q$ is a tautology. Two compound propositions that have the same truth values in all possible cases are called logically equivalent. The symbol $\equiv$ or $\iff$ is not a logical connective, and $p \equiv$ or $\iff q$ is not a compound proposition, but rather is the statement that $p \longleftrightarrow q$ is a tautology.*

A truth table with 2^n rows is needed to prove the equivalence of two compound propositions in n variables. Because 2^n grows extremely rapidly as n increases, the use of truth tables to establish equivalences becomes impractical as the number of variables grows. It is quicker to use logical equivalences to show the equivalence of two compound propositions. Also, we can reason about truth values.

Example 15 *Determine whether each of the compound propositions $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$, and $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is satisfiable.*

Solution: *Instead of using a truth table to solve this problem, we will reason about truth values. Note that $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ is true when the three variables p, q, and r have the same truth value (see Exercise 42 of Section 1.1). Hence, it is satisfiable as there is at least one assignment of truth values for p, q, and r that makes it true. Similarly, note that $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is true when at least one of p, q, and r is true and at least one is false (see Exercise 43 of Section 1.1). Hence, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is satisfiable, as there is at least one assignment of truth values for p, q, and r that makes it true.*

Finally, note that for $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ to be true, $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ and $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ must both be true. For the first to be true, the three variables must have the same truth values, and for the second to be true, at least one of the three variables must be true and at least one must be false. However, these conditions are contradictory. From these observations we conclude that no assignment of truth values to p, q, and r makes $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ true. Hence, it is unsatisfiable.

Logical Equivalence	Name
$p \vee F \equiv p$ $p \wedge T \equiv p$	Identity laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \vee q) \equiv \neg p \wedge \neg q$ $\neg(p \wedge q) \equiv \neg p \vee \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Negation laws
$p \to q \equiv \neg p \vee q$ $p \to q \equiv \neg q \to \neg p$ $p \vee q \equiv \neg p \to q$ $p \wedge q \equiv \neg(p \to \neg q)$ $\neg(p \to q) \equiv p \wedge \neg q$ $(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$ $(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$ $(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$ $(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$	Logical Equivalences involving Conditional Statements
$p \longleftrightarrow q \equiv (p \to q) \wedge (q \to p)$ $p \longleftrightarrow q \equiv \neg p \longleftrightarrow \neg q$ $p \longleftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$ $\neg(p \longleftrightarrow q) \equiv p \longleftrightarrow \neg q$	Logical Equivalences involving Biconditional Statements

Table 1.4: Logical Equivalences

1.2 Applications of Satisfiability

1.2.1 Translating Sentences

Translating sentences into a set of statements connected with logical operators removes the ambiguity. Moreover, once we have translated sentences into logical expressions, we can analyze these logical expressions to determine their truth values, we can manipulate them, and we can use rules of inference to reason about them.

Example 16 *How can this English sentence be translated into a logical expression?*

"You can access the Internet from campus only if you are a computer science major or you are not a freshman."

Solution: *Let a, c, and f represent "You can access the Internet from campus", "You are a computer science major", and "You are a freshman", respectively. Noting that "only if" is one way a conditional statement can be expressed, this sentence can be represented as* $a \to (c \vee \neg f)$.

Example 17 *How can this English sentence be translated into a logical expression?*

"You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old." [unless means if not]

Solution: *Let q, r, and s represent "You can ride the roller coaster", "You are under 4 feet tall", and "You are older than 16 years old", respectively. Then the sentence can be translated to* $(r \wedge \neg s) \to \neg q$.

1.2.2 Expressing System Specifications

System and software engineers take requirements in natural language and produce precise and unambiguous specifications that can be used as the basis for system development. System specifications should be consistent, that is, they should not contain conflicting requirements that could be used to derive a contradiction. When specifications are not consistent, there would be no way to develop a system that satisfies all specifications.

Example 18 *Determine whether these system specifications are consistent:*

"The diagnostic message is stored in the buffer or it is retransmitted".

"The diagnostic message is not stored in the buffer".

"If the diagnostic message is stored in the buffer, then it is retransmitted".

Solution: *Let* p *denote "The diagnostic message is stored in the buffer", and let* q *denote "The diagnostic message is retransmitted". The specifications can then be written as* $p \vee q$, $\neg p$, *and* $p \to q$. *An assignment of truth values that makes all three specifications true must have* p *false to make* $\neg p$ *true. Because we want* $p \vee q$ *to be true but* p *must be false, then* q *must be true. Again* $p \to q$ *is true, when* p *is false and* q *is true. Therefore, we conclude that these specifications are consistent, because they are all true when* p *is false and* q *is true.*

We could come to the same conclusion by use of a truth table to examine the four possible assignments of truth values to p *and* q.

1.2.3 Puzzle Solving

Example 19 *As a reward for saving his daughter from pirates, the King has given you the opportunity to win a treasure hidden inside one of three trunks. The two trunks that do not hold the treasure are empty. To win, you must select the correct trunk. Trunks 1 and 2 are each inscribed with the message "This trunk is empty", and Trunk 3 is inscribed with the message "The treasure is in Trunk 2". The Queen, who never lies, tells you that only one of these inscriptions is true, while the other two are wrong. Which trunk should you select to win?*

Solution: *Let* p_i *be the proposition that the treasure is in Trunk* i, *for* $i = 1$, *2, 3. To translate into propositional logic the Queen's statement that exactly one of the inscriptions is true, we observe that the inscriptions on Trunk 1,* *Trunk 2, and Trunk 3, are $\neg p_1$, $\neg p_2$, and p_2, respectively. So, her statement can be translated to*

$(\neg p_1 \wedge \neg(\neg p_2) \wedge \neg p_2) \vee (\neg(\neg p_1) \wedge \neg p_2 \wedge \neg p_2) \vee (\neg(\neg p_1) \wedge \neg(\neg p_2) \wedge p_2))$.

Using the rules for propositional logic,we see that this is equivalent to $(p_1 \wedge \neg p_2) \vee (p_1 \wedge p_2)$. By the distributive law, $(p_1 \wedge \neg p_2) \vee (p_1 \wedge p_2)$ is equivalent to $p_1 \wedge (\neg p_2 \vee p_2)$. But because $\neg p_2 \vee p_2$ must be true, this is then equivalent to $p_1 \wedge T$, which is in turn equivalent to p_1. So the treasure is in Trunk 1 (that is, p_1 is true), and p_2 and p_3 are false; and the inscription on Trunk 2 is the only true one.

Example 20 *An island has two kinds of inhabitants, knights, who always tell the truth, and their opposites, knaves, who always lie. You encounter two people A and B. What are A and B if A says "B is a knight" and B says "The two of us are opposite types"?*

Solution: *Let p and q be the statements that A is a knight and B is a knight, respectively, so that $\neg p$ and $\neg q$ are the statements that A is a knave and B is a knave, respectively.*

We first consider the possibility that A is a knight; this is the statement that p is true. If A is a knight, then he is telling the truth when he says that B is a knight, so that q is true, and A and B are the same type. However, if B is a knight, then B's statement that A and B are of opposite types, the statement $(p \wedge \neg q) \vee (\neg p \wedge q)$, would have to be true, which it is not, because A and B are both knights. Consequently, we can conclude that A is not a knight, that is, that p is false.

If A is a knave, then because everything a knave says is false, A's statement that B is a knight, that is, that q is true, is a lie. This means that q is false and B is also a knave. Furthermore, if B is a knave, then B's statement that A and B are opposite types is a lie, which is consistent with both A and B being knaves. We can conclude that both A and B are knaves.

Example 21 (Muddy Children Puzzle) *A father tells his two children, a boy and a girl, to play in their backyard without getting dirty. However,* *while playing, both children get mud on their foreheads. When the children stop playing, the father says "At least one of you has a muddy forehead," and then asks the children to answer "Yes" or "No" to the question: "Do you know whether you have a muddy forehead?" The father asks this question twice. What will the children answer each time this question is asked, assuming that a child can see whether his or her sibling has a muddy forehead, but cannot see his or her own forehead? Assume that both children are honest and that the children answer each question simultaneously.*

Solution: *Let s be the statement that the son has a muddy forehead and let d be the statement that the daughter has a muddy forehead. When the father says that at least one of the two children has a muddy forehead, he is stating that the disjunction $s \vee d$ is true. Both children will answer "No" the first time the question is asked, because each sees mud on the other child's forehead. That is, the son knows that d is true, but does not know whether s is true, and the daughter knows that s is true, but does not know whether d is true.*

After the son has answered "No" to the first question, the daughter can determine that d must be true. This follows because when the first question is asked, the son knows that $s \vee d$ is true, but cannot determine whether s is true. Using this information, the daughter can conclude that d must be true, for if d were false, the son could have reasoned that because $s \vee d$ is true, then s must be true, and he would have answered "Yes" to the first question. The son can reason in a similar way to determine that s must be true. It follows that both children answer "Yes" the second time the question is asked.

Example 22 (The n-Queens Problem) *The n-queens problem asks for a placement of n queens on an $n \times n$ chessboard so that no queen can attack another queen. This means that no two queens can be placed in the same row, in the same column, or on the same diagonal.*

Solution: *To model the n-queens problem as a satisfiability problem, we introduce n^2 variables, $p_{i,j}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$. For a given placement of a queens on the chessboard, $p_{i,j}$ is true, when there is a queen on the square in the i-th row and j-th column, and is false otherwise.*

For no two of the n queens to be in the same row, there must be one queen in each row. We can show that there is one queen in each row by verifying *that every row contains at least one queen (Q_1) and that every row contains at most one queen (Q_2).*

Every row contains at least one queen is asserted by

$$Q_1 = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} p_{i,j}$$

Every row contains at most one queen is asserted by

$$Q_2 = \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n-1} \bigwedge_{k=j+1}^{n} (\neg p_{i,j} \vee \neg p_{i,k})$$

Every column contains at most one queen is asserted by

$$Q_3 = \bigwedge_{j=1}^{n} \bigwedge_{i=1}^{n-1} \bigwedge_{k=i+1}^{n} (\neg p_{i,j} \vee \neg p_{k,j})$$

Note that the assertion Q_3 along with the previous assertion that every row contains a queen (Q_1 and Q_2), implies that every column contains a queen.

Note that squares (i_1, j_1) and (i_2, j_2) are on the same diagonal if either $i_1 + i_2 = j_1 + j_2$ or $i_1 - i_2 = j_1 - j_2$. To assert that no diagonal contains two queens, we assert

$$Q_4 = \bigwedge_{i=2}^{n} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{min(i-1,n-j)} (\neg p_{i,j} \vee \neg p_{i-k,k+j})$$

, and

$$Q_5 = \bigwedge_{i=1}^{n-1} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{min(n-i,n-j)} (\neg p_{i,j} \vee \neg p_{i+k,j+k})$$

The innermost conjunction in Q_4 and in Q_5 for a pair (i, j) runs through the positions on a diagonal that begin at (i, j) and runs rightward along this *diagonal. The upper limits on these innermost conjunctions identify the last cell in the board on each diagonal.*

Putting all this together, we find that the solutions of the n-queens problem are given by the assignments of truth values to the variables $p_{i,j}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$ that make $Q = Q_1 \wedge Q_2 \wedge Q_3 \wedge Q_4 \wedge Q_5$ true.

Example 23 (Sudoku) *Sudoku puzzles are constructed using a 9 × 9 grid, where each gid is made up of nine 3 × 3 subgrids, known as blocks. For each puzzle, some of the 81 cells, called givens, are assigned one of the numbers 1, 2, ..., 9, and the other cells are blank. The puzzle is solved by assigning a number to each blank cell so that every row, every column, and every one of the nine 3 × 3 blocks contains each of the nine possible numbers. Note that instead of using a 9 × 9 grid, Sudoku puzzles can be based on $n^2 \times n^2$ grids, for any positive integer n, with the $n^2 \times n^2$ grid made up of n^2 $(n \times n)$ subgrids.*

Sudoku puzzles have two additional important properties. First, they have exactly one solution. Second, they can be solved using reasoning alone, that is, without resorting to searching all possible assignments of numbers to the cells. As a Sudoku puzzle is solved, entries in blank cells are successively determined by already known values.

Solution: *To encode a Sudoku puzzle, let $p_{i,j,n}$ denote the propositional variable that is true when the number n is in the cell in the i-th row and j-th column. There are 9 × 9 × 9 = 729 such propositional variables, as i, j, and n all range from 1 to 9.*

Given a Sudoku puzzle, we begin by encoding each of the given values. Then, we construct compound propositions that assert that every row contains every number, every column contains every number, every 3 × 3 block contains every number, and each cell contains no more than one number.

Now, we will explain how to construct the assertion that every row contains every integer from 1 to 9 as follows.

- *For each cell with a given value, we assert $p_{i,j,n}$ when the cell in row i and column j has the given value n.*

- *We assert that every row contains every number:*

$$R_1 = \bigwedge_{i=1}^{9} \bigwedge_{n=1}^{9} \bigvee_{j=1}^{9} p_{i,j,n}$$

- *We assert that every column contains every number:*

$$R_2 = \bigwedge_{j=1}^{9} \bigwedge_{n=1}^{9} \bigvee_{i=1}^{9} p_{i,j,n}$$

- *We assert that each of the nine 3 $\times$ 3 blocks contains every number:*

$$R_3 = \bigwedge_{r=0}^{2} \bigwedge_{s=0}^{2} \bigwedge_{n=1}^{9} \bigvee_{i=1}^{3} \bigvee_{i=1}^{3} p_{3r+i,3s+j,n}$$

Now, we will explain how to construct the assertion that no cell contains more than one number, we take the conjunction over all values of n_1, n_2, i, and j, where each variable ranges from 1 to 9 and $n_1 \neq n_2$ of $p_{i,j,n_1} \to \neg p_{i,j,n_2}$.

Finally, the Sudoku puzzle is solved by finding an assignment of truth values to the 729 propositional variables $p_{i,j,n}$ with i, j, and n each ranging from 1 to 9 that makes the conjunction of all these compound propositions true.

1.3 Rules of Inference in Propositional Logic

Proof in mathematics is a valid argument that establishes the truth of mathematical statement. By an **argument**, we mean a sequence of statements that end with a conclusion. By **valid**, we mean that the conclusion, or final statement of the argument, must follow from the truth of the preceding statements, or **premises**, of the argument. To deduce a new statement from a set of given statements, we use **rules of inference** which are tools to construct a valid argument.

We can always use a truth table to show that an argument form is valid. We do this by showing that whenever the premises are true, the conclusion

Rule of Inference	Tautology	Name
p $p \to q$ ——— $\therefore q$	$(p \land (p \to q)) \to q$	Modus ponens or Law of detachment
$\neg q$ $p \to q$ ——— $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ ——— $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ ——— $\therefore q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p ——— $\therefore p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ ——— $\therefore p$	$(p \land q) \to p$	Simplification
p q ——— $\therefore p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ ——— $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

Table 1.5: Rules of Inference in Propositional Logic

must also be true. However, this can be a tedious approach. For example, when an argument form involves n different propositional variables, to use a truth table to show this argument form is valid requires 2^n different rows.

Fortunately, we do not have to resort to truth tables. Instead, we can first establish the validity of some relatively simple argument forms, called **rules of inference**. These rules of inference can be used as building blocks to construct more complicated valid argument forms.

Example 24 *Show that the premises "It is not sunny this afternoon and it is colder than yesterday", "We will go swimming only if it is sunny", "If we do not go swimming, then we will take a canoe trip", and "If we take a canoe trip, then we will be home by sunset" lead to the conclusion "We will be home by sunset".*

Solution: *Let p be the proposition "It is sunny this afternoon", q the proposition "It is colder than yesterday", r the proposition "We will go swimming", s the proposition "We will take a canoe trip", and t the proposition "We will be home by sunset". Then the premises become $\neg p \wedge q$, $r \to p$, $\neg r \to s$, and $s \to t$. The conclusion is simply t. We need to give a valid argument with premises $\neg p \wedge q$, $r \to p$, $\neg r \to s$, and $s \to t$ and conclusion t. We construct an argument to show that our premises lead to the desired conclusion as follows.*

Step	*Reason*
1. $\neg p \wedge q$	*Premise*
2. $\neg p$	*Simplification using (1)*
3. $r \to p$	*Premise*
4. $\neg r$	*Modus tollens using (2) and (3)*
5. $\neg r \to s$	*Premise*
6. s	*Modus ponens using (4) and (5)*
7. $s \to t$	*Premise*
8. t	*Modus ponens using (6) and (7)*

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Example 25 *Show that the premises "If you send me an e-mail message, then I will finish writing the program", "If you do not send me an e-mail message, then I will go to sleep early", and "If I go to sleep early, then I will* *wake up feeling refreshed" lead to the conclusion "If I do not finish writing the program, then I will wake up feeling refreshed".*

Solution: *Let p be the proposition "You send me an e-mail message", q the proposition "I will finish writing the program", r the proposition "I will go to sleep early", and s the proposition "I will wake up feeling refreshed". Then the premises are $p \to q$, $\neg p \to r$, and $r \to s$. The desired conclusion is $\neg q \to s$. We need to give a valid argument with premises $p \to q$, $\neg p \to r$, and $r \to s$ and conclusion $\neg q \to s$. This argument form shows that the premises lead to the desired conclusion.*

Step	*Reason*
1. $p \to q$	*Premise*
2. $\neg q \to \neg p$	*Contrapositive of (1)*
3. $\neg p \to r$	*Premise*
4. $\neg q \to r$	*Hypothetical syllogism using (2) and (3)*
5. $r \to s$	*Premise*
6. $\neg q \to s$	*Hypothetical syllogism using (4) and (5)*

Example 26 *Use resolution to show that the hypotheses "Jasmine is skiing or it is not snowing" and "It is snowing or Bart is playing hockey" imply that "Jasmine is skiing or Bart is playing hockey".*

Solution: *Let p be the proposition "It is snowing", q the proposition "Jasmine is skiing", and r the proposition "Bart is playing hockey". We can represent the hypotheses as $\neg p \vee q$ and $p \vee r$, respectively. Using resolution, the proposition $q \vee r$, "Jasmine is skiing or Bart is playing hockey", follows.*

Example 27 *Show that the premises $(p \wedge q) \vee r$ and $r \to s$ imply the conclusion $p \vee s$.*

Solution: *We can rewrite the premises $(p \wedge q) \vee r$ as two clauses, $p \vee r$ and $q \vee r$. We can also replace $r \to s$ by the equivalent clause $\neg r \vee s$. Using the two clauses $p \vee r$ and $\neg r \vee s$, we can use resolution to conclude $p \vee s$.*