

Chapter 2

Logic and Reasoning Patterns



Abstract Logic is the foundation of AI, and the majority of AI's principles are based on logical or deductive reasoning. The chapter presents: contributions of pioneers of logic, the argumentation theory, which is based on logic and with its roots in propositional logic, the process of validating the propositional formulas, their syntax and semantics, interpretation of a logical expression through semantic tableau, followed with presents the basic reasoning patterns used by human, and their formal notations. In addition, presents the normal forms of propositional formulas and application of resolution principle on these for inference. The nonmonotonic reasoning and its significance is briefly described. At the end, the chapter presents the axiomatic system due to Hilbert and its limitations, and concludes with chapter summary.

Keywords Logic · Propositional logic · Deductive reasoning · Argumentation theory · Syntax and semantics of propositional formulas · Nonmonotonic reasoning · Hilbert's axiomatic system

2.1 Introduction

The ancient *Greeks* are the source of modern logic, their education system emphasized the competence in rhetoric (proficient in language) and philosophy; the words *axioms* and *theorem* are from Greek. The logic was used to formalize the deductions—the derivation of true *conclusions*—from true *premises*. Later it was formalized as a set theory by the mathematician *George Boole*. Till the arrival of the nineteenth century, the logic remained more of a philosophical nature, rather than a mathematical and scientific tool. Later, since complex things could not be reasoned through logic, the logic became part of mathematics, where mathematical deduction became justifiable through formalizing a system of logic, and resulted in one very important breakthrough. This was, about the set of true statements, stated as “the set of provable statements are only those that are true statements.” This is because some proof exists for those due to some other true statements.

At the beginning of nineteenth century, the mathematician *David Hilbert* introduced the logic, as well as theories of the nature of logic—a far more generalization

of the logic. But, this generalization received a blow when another mathematician *Kurt Gödel* showed in 1931 that there are true statements of arithmetics that are not provable, through his *incompleteness* theorem.

Now, though mathematical logic remains the branch of pure mathematics, it is extensively applied to computer science and artificial intelligence in the form of propositional logic and predicate logic (first-order predicate logic (FOPL)).

As per the Newell's and Simons's Physical Symbol System Hypothesis (PSSH), discussed in the previous chapter, the knowledge representation is the first requirement of achieving intelligence. This chapter presents the knowledge representation using *propositional logic*, introduces first-order predicate logic (FOPL), and drawing of inferences using propositional logic.

Logic is a formal method for reasoning, using its concepts can be translated into symbolic representation, which closely approximate the meaning of these concepts. The symbolic structures can be manipulated using computer programs to deduce facts to carry out the form of automated reasoning [9].

The *aim* of logic is to learn principles of valid reasoning as well as to discern good reasoning from bad reasoning, identifying invalid arguments, distinguishing *inductive* versus *deductive* arguments, identifying *fallacies* as well as avoiding the fallacies.

The *Objective* of logic is to equip oneself with various tools and techniques, i.e., *decision procedures* for validating given arguments, detecting and avoiding fallacies of a given deductive or inductive argument.

We study the logic because of the following reasons:

- Logic deals with what follows from what? For example, Logical consequence, inference pattern, and validating such patterns,
- We want the computer to understand our language and does some intelligent tasks for us (Knowledge representation),
- To engage in debates, solving puzzles, game like situation,
- Identify which one is a fallacious argument and what is a type of fallacy?
- Proving theorems through *deduction*. To find out whether whatever proved is correct, or whatever obviously true has a proof, and
- To solve some problems concerning the foundations of mathematics.

Learning Outcomes of This Chapter:

1. Convert logical statements from informal language to propositional logic expressions. [Usage]
2. Apply formal methods of symbolic propositional such as calculating the validity of formula and computing normal forms. [Usage]
3. Use the rules of inference to construct proofs in propositional. [Usage]
4. Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as software analysis (e.g., program correctness), database queries, and algorithms. [Usage]

5. Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles. [Usage]
6. Explain the difference between rule-based and model-based reasoning techniques. [Familiarity]
7. Describe the strengths and limitations of propositional logic. [Familiarity]

2.2 Argumentation Theory

The *Argumentation theory* is the study of how conclusions can be reached through logical reasoning, that is, whether the claims are soundly based on premises or not (Fig. 2.1). It includes the arts and sciences of civil debate, dialog, conversation, and persuasion. It includes the studies of rules of inference, logic, and procedural rules in both artificial and real world settings.

An argumentation system comprises debate and negotiations, aimed at reaching to a mutually agreeable conclusion. The argumentation may also consist of erroneous dialogs, where victory over an opponent is the only goal, without consideration of the truth. The argumentation theory is an art, as well as science, using these people protect their self-interests and beliefs using rational dialogs at commonplaces of their meeting points, and during the process of argumentation.

People make use of argumentation theory in law also, for example, in preparing an argument to be presented before the court of law, in debate in the court of law, in trials, and in testing the validity of certain kinds of evidence. Scholars of Argumentation theory study the post-hoc rationalizations by which organizational actors try to justify decisions even made irrationally.

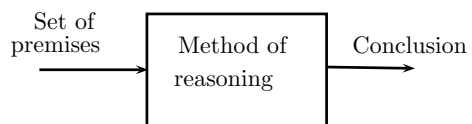
The simple block diagram for logical reasoning shown in Fig. 2.1 has internal structure, comprising the following:

1. a set of assumptions or premises (or antecedents),
2. a method of reasoning or deduction, and
3. a conclusion or consequence.

If the premises are P_1, P_2, \dots, P_n , then they are conjuncted and their conjunction imply the conclusion C , i.e., $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow C$.

An argument must have at least one premise and one conclusion. Often, classical logic is used as the method of reasoning so that the conclusion follows logically from the assumptions or support. One challenge is that if the set of assumptions is

Fig. 2.1 Inference process



inconsistent then anything can follow logically from inconsistency. Therefore, it is common to insist that the set of assumptions be consistent. It is the practice to have a minimal set. Such argumentation has been applied to the field of medicine also.

The second school of argumentation investigates abstract arguments, where “argument” is considered a primitive term, so no internal structure of arguments is taken on the account.

2.3 Role of Knowledge

We have discussed in the above section about knowledge and logic. The Logic needs a base of knowledge to infer or conclude new knowledge. The knowledge is also used for *learning*, *retrieval* and *reasoning*. The learning is not only adding new facts into an existing knowledge base, but before the new data are put into the storage, they need to be classified for ease of retrieval. The interaction and inference with existing facts avoid the redundancy and duplication of knowledge in the knowledge base. In addition, the learning updates the existing facts.

Having stored the knowledge in the process of learning, one important objective of that is, retrieval. The representation scheme used in the knowledge base has critical effect on the efficiency of the retrieval system. As humans, we are very good in retrieval from our knowledge (memories), and many AI systems have used that for modeling AI learnings.

The knowledge is also used for reasoning process, i.e., to infer new facts from the existing facts in the knowledge. For example, observing many birds flying, to infer that all the birds fly, as well for solving a complex problem, say, based on sufficient facts, to infer that a customer financed by a bank, will be able to repay the loan the bank has financed to him.

2.4 Propositional Logic

The propositional logic deals with individual Propositions, which are viewed as *atoms*, i.e., these cannot be further broken down into smaller constituents. For building propositional logic, first we describe the logic with the help of a formula called *Well-Formed Formulas* (wff, read as woofs). A formula is a *syntactic* concept, which means whether or not a string of symbols is a formula not. It can be determined solely based on its formal construction, i.e., whether it can be built according to its construction rules. Therefore, we are in a position to verify that a sequence of symbols is a formula or not, as per the specified rules. This function of verification, in a compiler, is done by a *parser*—to verify whether the formula belongs to the particular programming language or not. A parser also constructs a parse-tree of the given formula through which it tells how the formula is constructed [2].

The meaning (*semantics*) is associated with each formula by defining its *interpretation*, which assign a value *true* (T) or *false* (F) to every formula. The syntax is also used to define the concept of *proof*—the symbolic manipulations of formulas to deduce the given theorem. The important thing we should note is that provable formulas are only those which are always true.

We start the propositional logic with the individual propositional variables. These variables themselves are formulas, which cannot be further analyzed. We represent these by English alphabets and subscripted alphabets $p, q, r, s, t, p_1, p_2, q_1, q_2, \dots$, etc. These formulas may have smaller constituents but it is not the role of propositional logic to go into the details of their constructions. The use of letters to represent propositions is not in true sense variables, they simply represent the propositions or statements in a symbolic form, and they are not the variables in the sense used in *predicate logic* (to be discussed later), or in *high-level languages* like *C* or *Fortran*, where a variable stands for a domain of values. For example, an integer variable in a Fortran program stands for any integer number as per the specifications of the language.

The other symbols of propositional logic are *operators* as follows:

- \wedge conjunction operator,
- \vee disjunction operator,
- \neg not or inverting operator,
- \rightarrow implication, i.e., if ... than ... rule, and
- \perp contradiction (false).

Let following be the propositions:

p = Sun is star.

q = Moon is satellite.

We can construct the following formulas using the above propositions:

$p \wedge q$ = Sun is star *and* Moon is satellite.

$p \vee q$ = Sun is star *or* Moon is satellite tennis.

$\neg p \vee q$ = Sun is *not* star or Moon is satellite.

$\neg p \rightarrow q$ = *if* Sun is *not* star *then* Moon is satellite.

A formula in propositional logic can be recursively defined as follows:

- (i) Each propositional variable and null are formulas, therefore, p, q, ϕ are formulas,
- (ii) If p, q are formulas, then $p \wedge q, p \vee q, \neg p, p \rightarrow q, (p)$, are also formulas,
- (iii) A string of symbols is a formula only as determined by finitely many applications of above (i) and (ii), and
- (iv) nothing else is propositional formula.

This recursive form of the definition can be expressed using *BNF* (Backups-Naur Form) notation as follows:

1. $formula := atomicformula \mid formula \wedge formula \mid formula \vee formula$
 $\mid formula \rightarrow formula \mid \neg formula \mid (formula)$
 2. $atomicformula := \perp \mid p \mid q \mid r \mid p_0 \mid p_1 \mid p_2 \mid \dots$
- (2.1)

In the above notation, the symbols—*formula* and *atomicformula*, that appears to the left-hand are called *non-terminals* and represent grammatical classes. The p, q, r, \perp, p_1 , etc, that appear only to the right-hand side, are called *terminals*, and represent the symbols of the language.

A sentence in the propositional language is obtained through a derivation that starts with a non-terminal, and repeatedly applied the substitution rules from the BNF notations, until the terminals are reached [8].

Example 2.1 Derivation for $p \wedge q \rightarrow r$.

The sequence of substitutions rules to derive this formula, i.e., to establish that it is syntactically correct, are as follows:

$$\begin{aligned}
 formula &\Rightarrow formula \rightarrow formula \\
 &\Rightarrow formula \wedge formula \rightarrow formula \\
 &\Rightarrow atomic \wedge formula \rightarrow formula \\
 &\Rightarrow p \wedge formula \rightarrow formula \\
 &\Rightarrow p \wedge atomic \rightarrow formula \\
 &\Rightarrow p \wedge q \rightarrow formula \\
 &\Rightarrow p \wedge q \rightarrow atomic \\
 &\Rightarrow p \wedge q \rightarrow r.
 \end{aligned}$$

The symbol *atomic* stands for atomic formula and the symbol “ \Rightarrow ” stands for “implies”, i.e., the expression to right to this is implied by the expression to left of “ \Rightarrow ”.

The derivation can also be represented by a *derivation-tree* (*parse-tree*), shown in Fig. 2.2. From the derivation-tree, we can obtain another tree shown in Fig. 2.3, called *syntax-tree* or *formation-tree*, by replacing each non-terminal by the child that is an operator under that. There is always unique syntax-tree for every formula. \square

Considering two propositions p, q , the interpretation (semantics) of the formulas constructed when they are joined using binary operators ($\vee, \wedge, \rightarrow$) are shown in the truth-table Table 2.1.

The *Material conditional* ‘ \rightarrow ’ joins two simpler propositions, e.g., $p \rightarrow q$, read as “if p then q ”. The proposition to the left of the arrow is called the *antecedent* and to the right is *consequent*. There is no such designation for conjunction or disjunction operators because they are commutative operations. The $p \rightarrow q$ expresses that q is true whenever p is true. Thus it is true in every case in Table 2.1, except in row three, because this is the only case when p is true but q is not. Using “if p then q ”,

Fig. 2.2 Parse-tree for the expression $p \wedge q \rightarrow r$

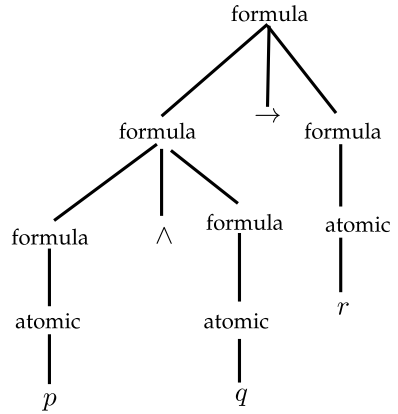


Fig. 2.3 Syntax-tree for the expression $p \wedge q \rightarrow r$

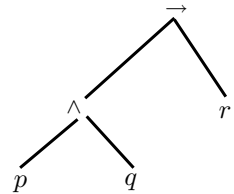


Table 2.1 Interpretation of propositional formulas

p	q	$p \vee q$	$p \wedge q$	$p \rightarrow q$
F	F	F	F	T
F	T	T	F	T
T	F	T	F	F
T	T	T	T	T

we can express that “if it is raining outside then there is a cold over Kashmir”. The material conditional is often confused with *physical causation*. The material conditional, however, only relates two propositions by their truth values—which is not the relation of *cause* and *effect*. It is contentious in the literature whether the material implication represents logical causation.

2.4.1 Interpretation of Formulas

The interpretation of formula is assigning truth value to that formula. As discussed earlier, a formula can be atomic or in may be complex, i.e., joining or atomic formulas. The following are some definitions related to the interpretation of formulas [1].

Definition 2.1 (*Satisfied, model, valid, and tautology*) A propositional formula A is *satisfied* iff $\mathcal{I}(A) = \text{True}$ for some interpretation \mathcal{I} . A *satisfying interpretation* is called *model* for A . The formula A is called *valid*, denoted by $\models A$, iff $\mathcal{I}(A) = \text{True}$ for all interpretations \mathcal{I} . A valid propositional formula is also called *tautology*.

A propositional formula is *unsatisfiable* (also called *contradiction*, \perp), iff it is not satisfiable, i.e., $\mathcal{I}(A) = \text{False}$, for all interpretations \mathcal{I} . If $\mathcal{I}(A) = \text{False}$ for some interpretation \mathcal{I} , then A is called *non-valid* or *falsifiable*, and denoted by $\not\models A$.

Definition 2.2 (*Simultaneously satisfiable*) A set of formulas $S = \{A_1, A_2, \dots, A_n\}$ is *simultaneously satisfiable* iff there exists an interpretation \mathcal{I} such that $\mathcal{I}(A_i) = \text{True}$ for all i . The S is *unsatisfiable* iff for every interpretation \mathcal{I} there exists an i such that $\mathcal{I}(A_i) = \text{False}$.

2.4.2 Logical Consequence

The *logical consequence* or *logically follows* is the central concept in the foundations of logic. It is much more interesting to assume that a set of formulas is true and then to investigate the consequences of these assumptions [1].

Assume that θ and ψ are formulas (sentences) of a set \mathcal{P} , and \mathcal{I} is an interpretation of \mathcal{P} . The sentence θ of propositional logic is true under an interpretation \mathcal{I} iff \mathcal{I} assigns the truth value T to that sentence. The θ is false under an interpretation \mathcal{I} iff θ is not true under \mathcal{I} .

Definition 2.3 (*Logical consequence*) A sentence ψ of propositional logic is a *logical consequence* of a sentence (or set of sentences) θ , represented as $\theta \models \psi$, if every interpretation \mathcal{I} that satisfy θ also satisfy ψ .

In fact, ψ need not be true in every possible interpretation, only in those interpretations which satisfy θ , i.e, those interpretations which satisfy every formula in θ . In the formula $((p \rightarrow q) \wedge p) \vdash q$, the q is logical consequence of $((p \rightarrow q) \wedge p)$. The sign ' \vdash ', is sign of *deduction*, and $S \vdash q$ is read as S deduces q , where S is a set of formulas and q is the formula.

A sentence of propositional logic is *consistent* iff it is true under at least one interpretation. It is *inconsistent* if it is not consistent.

Example 2.2 Determine the logical consequence of $\psi = (p \vee r) \wedge (\neg q \vee \neg r)$ from $\theta = \{p, \neg q\}$, i.e., find $\theta \models \psi$, and validity for ψ .

Here ψ is logical consequence of θ , denoted by $\theta \models \psi$, because ψ is true under all the interpretations such that $\mathcal{I}(p) = \text{True}$, and $\mathcal{I}(q) = \text{False}$, is the interpretation, for which θ is satisfied.

However, ψ is not valid, since it is not true under the interpretation $\mathcal{I}(p) = F$, $\mathcal{I}(q) = T$, $\mathcal{I}(r) = T$.

Further note that $\theta \vdash \psi$ is a valid statement because the expression $\theta \vdash \psi$ is always true. \square

2.4.3 Syntax and Semantics of an Expression

Syntax is name given to a correct *structure* of a statement. It is the meaning associated with the expression. It is mapping to the real-world situation is *semantics*. The semantics of a language defines the truth of each sentence with respect to each possible *world*. For example, the usual semantics for interpretation of the statement $(p \vee q) \wedge r$ is true in a *world* where either p or q or both are *true* and r is *true*. Different worlds can be all the possible sets of *truth* values of p, q, r , which is total 8. The truth values are simply the assignment to these variables, and not necessarily the values which are only *true*. For example, $\mathcal{I}(p) = F, \mathcal{I}(q) = F, \mathcal{I}(r) = T$; and $\mathcal{I}(p) = T, \mathcal{I}(q) = F, \mathcal{I}(r) = T$ are the possible worlds for the expression $(p \vee q) \wedge r$.

2.4.4 Semantic Tableau

Semantic tableau is relatively efficient method for deciding satisfiability for the formula of propositional calculus. The method (or algorithm) *systematically* searches for a model for a formula. If it is found, the formula is satisfiable, else not satisfiable. We start with the definition of some terms, and then analyze some formulas to motivate us for the construction of semantic tableau [1].

Definition 2.4 (*Literal and complementary pair*) A literal is an atom or negation of an atom. For any atom p , the set $\{p, \neg p\}$ is called complementary pair of literals. For any formula A , $\{A, \neg A\}$ is complementary pair of formulas.

Example 2.3 Analysis of the satisfiability of a formula.

Consider that a formula $A = p \wedge (\neg q \vee \neg p)$, has an arbitrary interpretation \mathcal{I} . Given this, $\mathcal{I}(A) = T$ iff $\mathcal{I}(p) = T$ and $\mathcal{I}(\neg q \vee \neg p) = T$. Hence, $\mathcal{I}(A) = T$ iff either,

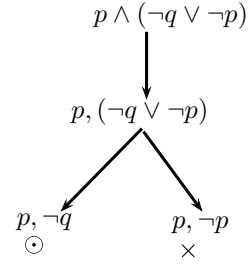
1. $\mathcal{I}(p) = T$ and $\mathcal{I}(\neg q) = T$, or
2. $\mathcal{I}(p) = T$ and $\mathcal{I}(\neg p) = T$.

Hence A is satisfiable if either (1) interpretation holds or (2) holds. But (2) is not feasible. So, A is satisfiable when the interpretation of (1) holds true. Note that the satisfiability of a formula is reduced to the satisfiability of literals.

It is clear that a set of literals is satisfied if and only if it does not contain complementary pair of literals. In the above case, the pair of literals $\{p, \neg p\}$ in case (2) is complementary pair, hence the formula is unsatisfied for this interpretation. However, the first set $\{p, \neg q\}$ is not the complementary pair, hence it is satisfiable.

From the above discussion, we have trivially constructed a model for the formula A by assigning *True* to positive literals and *False* to negative literals. Hence, $p =$

Fig. 2.4 Tree for semantic tableau



True, and $q = \text{False}$ makes the set in (1) true, hence $\{p = T, q = F\}$ is a model for formula A .

The above is a *search process*, and can be represented by a tree shown in Fig. 2.4. The leaves in the tree represent a set of literals that must be satisfied. A leaf containing complementary pair of literals is marked *closed* by \times , while the satisfying leaf is marked as *open* by \odot .

The construction process of the tree can be represented as an algorithm, to find out if some model exists for a formula, and what is that model. \square

Definition 2.5 (*Semantic Tableau*) Semantic Tableau is a tree, each node of which will be labeled with a set of formulas, and these formulas are inductively expanded to leaves such that each leaf is marked as open by \odot or closed by \times .

Definition 2.6 (*Completed tableau*) A semantic tableau whose construction is terminated is called completed tableau. A completed tableau is closed if all the leaves are marked closed. Otherwise, it is open i.e., some leaves are open.

Definition 2.7 (*Unsatisfiable formula*) Any formula A is unsatisfiable if its completed tableau \mathcal{T} is closed.

Corollary 2.1 (Method for semantic tableau) A formula A is satisfied if its tableau \mathcal{T} is open. Thus a method for semantic tableau is an algorithm for the validity of a propositional calculus formula.

Example 2.4 Find out whether $(p \vee q) \wedge (\neg p \wedge \neg q)$ is satisfiable, using tableau method.

Let $A = (p \vee q) \wedge (\neg p \wedge \neg q)$. For the satisfaction of A , $\mathcal{I}(A) = \text{True}$ for some assignments. That is, $\mathcal{I}(p \vee q) = \text{True}$ and $\mathcal{I}(\neg p \wedge \neg q) = \text{True}$. Thus, $\mathcal{I}(A)$ is *True* if either,

- $\mathcal{I}(p) = T, \mathcal{I}(\neg p) = \text{True}, \mathcal{I}(\neg q) = \text{True}$, or
- $\mathcal{I}(q) = \text{True}, \mathcal{I}(\neg p) = \text{True}, \mathcal{I}(\neg q) = \text{True}$.

So that, two sets of literals are,

$(p, \neg p, \neg q)$ and $(q, \neg p, \neg q)$.

Since both contain complementary pairs, hence neither of the literals is satisfiable. So it is impossible to find a model for A , and A is unsatisfiable.

2.5 Reasoning Patterns

How can we reason about solving any problem? To a certain extent, it depends on the chosen knowledge representation. The followings are the methods in broad about how the reasoning is performed by humans [2].

Deductive Reasoning

It is a process by which general premises are used to obtain the inferences, which are specific. For example, we may have the following premises and conclusion:

Premise-I: I do shopping when the weather is good on weekends.

Premise-II: Today is Saturday and the sky is clear.

Conclusion: Therefore, I will go for shopping Today.

To perform the deductive reasoning, the problem is first formulated in the way as we did in the above example. Having done this, the conclusions must be valid when the premises are true. Beginning with a small set of *axioms, postulates, and definitions*, the Greek mathematician Euclid proved a total of 465 geometric propositions as the logical consequences of the input assumptions.

One of the most fundamental rules of inference is *modus ponens* rule. We have the following example for modus ponens.

Premise-I: All the men are mortal.

Premise-II: Socrates is man.

Conclusion: Therefore, Socrates is mortal.

The new knowledge, “Socrates is mortal” has been deduced from the earlier two sentences.

The enumeration table of all possible worlds for *modus ponens* are shown in Table 2.2. We note that it is a valid inference, as the sentence $((p \rightarrow q) \wedge p) \rightarrow q$, with q as the inference implied, is true in all the rows.

Other deductive reasoning approaches are : *modus tollens* and *sylogism*, and *abduction*. The Table 2.3 shows the formulas for these rules.

Abduction is deductive type logic, which provides only a “plausible inference.” For example, given that: “smoking causes lung cancer” and “Sam died due to lung cancer”, through abduction one would infer that “Sam was smoker”. However, this

Table 2.2 Modus ponens is valid inference

p	q	$p \rightarrow q$	$(p \rightarrow q) \wedge p$	$((p \rightarrow q) \wedge p) \rightarrow q$
F	F	T	F	T
F	T	T	F	T
T	F	F	F	T
T	T	T	T	T

Table 2.3 Inference rules

Rule	Formula	Description
Modus ponens	$((p \rightarrow q) \wedge p) \vdash q$	If p then q ; p ; therefore q
Modus tollens	$((p \rightarrow q) \wedge \neg q) \vdash \neg p$	If p then q ; not q ; therefore not p
Abduction	$(p \rightarrow q) \wedge q \vdash p$	if p then q ; q ; therefore q
Hypothetical syllogism	$((p \rightarrow q) \wedge (q \rightarrow r))$	if p then q ; $\vdash (p \rightarrow r)$ if q then r ; therefore, if p then r
Disjunctive syllogism	$((p \vee q) \wedge \neg p) \vdash q$	Either p or q , or both; not p ; therefore, q

conclusion is not necessarily true, because there are other reasons also for lung cancer, which are not due to smoking. When statistics and probability theory are used along with abduction, it may result in most probable inferences out of the many likely inferences. To illustrate how the abduction based reasoning works, we consider a logical system comprising a general rule and one specific proposition.

All successful enterprising industrialists are rich (general rule). Rajan is a rich person (specific proposition). Therefore, a plausible inference can be that Rajan is a successful, enterprising industrialist.

However, this conclusion can be false also, because there are many other paths to richness, such as a lottery, inherited property, coming across a treasure, and so on. If we have a table of all the riches and how they became rich, we may draw the probability of abduction for richness to be true in this case.

Inductive Reasoning

The inductive reasoning arrives at a conclusion about all members of a class. It is based on examination of only a few members of the class and based on that it generalizes for the entire class. It is broadly reasoning from a *specific* to the *general*. For example, the traffic police comes to know about following situation on a particular day about nature of road accidents:

1st accident was due to wrong side drive,
2nd accident was due to wrong side drive,
3rd accident was due to wrong side drive.

One would logically infer that all the accidents are due to wrong side driving. Another example is about the birds for their flying attribute.

Crow fly,
peacock fly,
pigeon fly.

Thus, we conclude that all the birds fly.

Another example is about the progressive sum of 1st n odd integers:

$$1 = 1^2$$

$$1 + 3 = 2^2$$

$$1 + 3 + 5 = 3^2$$

$$1 + 3 + 5 + 7 = 4^2$$

Thus, by induction we prove that, the sum of n successive odd integers is n^2 .

The outcome of the inductive reasoning process will frequently contain some measures of uncertainty because including all possible facts in the premises are usually impossible.

We know that the inference of an accident's example is not always true, and also of "all birds fly" is not true, because, ostrich and penguins do not fly. However, for 1st odd integers sum, it is true.

The deductive or inductive approaches are used in logic, rule-based systems, and in frames.

Analogical Reasoning

The analogical reasoning assumes that when question is asked, the answer can be derived by analogy, as in the case of following example.

Premise: All the 100m racers get 5% additional in their merit score.

Question: How much one 400m racer will get additional in academic score?

Conclusion: Because, 400m is a race, and an sports activity like 100m, so it will also benefit one with 5% in final scores.

Analogical reasoning is a type of verbalization of an internalized learning process. An individual uses processes that require the ability to recognize previously encountered experiences. This approach is not very common in AI, however, the case-based reasoning, semantic networks, and frames use this analogical reasoning approach.

Formal Reasoning

It uses the process of syntactic manipulation of data structures to deduce new facts. A typical example is the mathematical logic used in proving theorems in geometry. For example, proof by *resolution*.

Procedural and Numeric Reasoning

It uses mathematical models or simulation to solve the problems. The model-based reasoning is an example of this approach.

Generalization and Abstraction

The approaches of generalization and abstraction, both can be used with the logical and semantic representation of knowledge.

Meta-level Reasoning

The meta-level reasoning involves the knowledge about what you, how much you know about so and so. Also, which approach to use, how successful the inference will

be, depends on a great extent on which knowledge representation method is used. For example, reasoning by analogy can be more successful with semantic networks than with frames.

2.5.1 Rule-Based Reasoning

The rule-based reasoning is also called *pattern matching*, and uses forward and backward chaining. The implementation of rule-based system makes use of modus ponens and other approaches. Consider the rule:

Rule 1: If export rises the prosperity increases.

Using the modus ponens, if the premises, e.g., “The export rises” is *true*, the conclusion of the rule is accepted as *true*. We call this accepting the rule as “rule fires”. The firing of a rule occurs when all its premises are satisfied, whether all are true or some are false. On firing, the resulting conclusion is stored in the assertion base, to use for further firing of the rules and generate the assertions. When a premise is not available as an assertion, it can be obtained by querying the user, or by firing other rules. Testing of a rule premise or conclusion is as simple as matching a symbol pattern.

Every rule in the knowledge base can be checked to see if its premises or conclusion can be satisfied by previously made assertions. This process of matching, if done using forward chaining, i.e., premises to conclusions. If it is done from conclusions to premises, it is called *backward chaining*.

2.5.2 Model-Based Reasoning

A reasoning within a context is important in any reasoning system. In real-life situations, one often provides a lot of missing contexts or out of context information when answering certain queries. This situation can be correctly modeled by supplementing the existing knowledge about the world, with additional context-specific information. When it is supplemented by context information, reasoning within context becomes a deduction process.

The added information may act as constrain to the existing information in the system, as in the absence of this additional information the deduction process has more paths of freedom in the reasoning process. But, due to the availability of this added context information the reasoning task becomes easier because the domain in which reasoning takes place gets restricted (constrained) due to having lesser flexibility of deduction paths to be navigated. This task can be formalized as a task of varying contexts.

The knowledge that comprises the information for reasoning in the model-based system is in the form of a set of models of the world. These models satisfy the

assignments and examples of the world. This is, in contrast, to the use of only the formulas in the first-order predicate logic to describe the world. The other difference is that the model-based approach is motivated from a cognitive point of view – the forerunners of this approach of reasoning are cognitive psychologists who support the “reasoning by examples.” When a model-based reasoning system is presented with a query, the reasoning is performed by evaluating the query on these models.

Let us suppose that model-based knowledge base representation Γ , and a query α are both given, and it is required to find out if Γ implies α (i.e., $\Gamma \models \alpha$)? This we can determine in two steps: 1) evaluate α on all the models in the representation, 2) If there is a model of Γ that does not satisfy α , the Γ does not model the alpha (i.e., $\Gamma \not\models \alpha$), otherwise we conclude that $\Gamma \models \alpha$. This means if the model-based representation contains all the models of Γ , then by definition, this approach verifies the implication correctly, and produces the correct deduction.

However, there is a problem—the representation of Γ , such that it explicitly holds all the models, is not a plausible solution. The model-based approach is feasible only if Γ can be replaced by small model-based representation, and after that also it should correctly support the deduction.

Various topics in reasoning are as follows:

- Monotonic versus nonmonotonic reasoning,
- Reasoning with uncertainty,
- Shallow and deep representation of knowledge,
- Semantic networks,
- Blackboard approach,
- Inheritance approach,
- Pattern matching,
- Conflict resolution.

These are discussed in current, and the following chapters, in details.

2.6 Proof Methods

There are two different methods, one is through *model checking* and other is *deduction* based. The first comprises enumeration of truth-tables, and is always exponential in n , where n is the size of the set of propositional symbols. The other, i.e., deduction based approach is repeated application of inference rules. The inference rules are used as operators in the standard search algorithm. In fact, the application of the inference approach to proof is called searching for solution. Proper selection of search directions is important here, as these will eliminate many unnecessary paths that are not likely to result in the goal. Consequently, the proof-based approach for reasoning is considered better and efficient compared to model enumeration/checking based method. The later is exhaustive and exponential in n , where n is the size of the set of propositional symbols.

The property, the logical system follows, is the fundamental property of *monotonicity*. As per this, if $S \vdash \alpha$, and β is additional assertion, then $S \wedge \beta \vdash \alpha$.

Thereby, the application of inference rules is legitimate (*sound*) rule, which helps in the generation of new knowledge from the existing. If a search algorithm like DFS (depth first search) is used, it will always be possible to find the proof, as it will search the goal, whatever the depth may it be. Hence, the inference method in this case is *complete* also [7].

Before the inference rules are applied on the knowledge base, the existing sentences in the knowledge base (KB) needs to be converted into some *normal form*.

2.6.1 Normal Forms

A logical expression can be represented as sum-of-product terms or product-of-sum terms. If a given logical expression is represented as sums of elementary products, then this form is called *disjunctive normal form* (DNF), and if it is represented as product of elementary sums, it is called *conjunctive normal form* (CNF). In DNF, the elementary product terms are called *minterms*, while in a CNF elementary sum terms are called *maxterms*. For a given formula, an equivalent disjunctive normal form with only disjunctions of minterms is called *principle disjunctive normal form* or *sum-of-products canonical form*. Similarly, an equivalent CNF with only conjunctions of maxterms is called *principle conjunctive normal form* or *product-of-sums canonical form* [2].

One technique to get a CNF expression for a given DNF expression, say, $\neg a \neg bc + \neg ab \neg c + \neg abc + a \neg bc$ is given in steps as follows:

1. Considering a DNF expression of three variable a, b, c , write down all the minterms: $\neg a \neg b \neg c, \neg a \neg bc, \neg ab \neg c, \neg abc, a \neg b \neg c, a \neg bc, ab \neg c, abc$.
2. Cross out all combinations in the original DNF. We are left with $\neg a \neg b \neg c, a \neg b \neg c, ab \neg c, abc$.
3. Next, write the expression in CNF by inverting each subset of three variables and ORing as $(a + b + c)(\neg a + b + c)(\neg a + \neg b + c)(\neg a + \neg b + \neg c)$ in the form of CNF.

Obtaining DNF from CNF is just the reverse process.

2.6.2 Resolution

The *resolution rule* is an inference which uses *deduction* approach. It is used in theorem proving. If two disjunctions have complementary literals, then a resultant inference of these is disjunction of these expressions, with complementary terms removed. If $p = p_1 \vee p_2 \vee c$ and $q = q_1 \vee \neg c$ are two formulas, then resolution of

p and q results to dropping of c and $\neg c$ and disjunction is performed of the remaining propositions of p and q , as follows:

$$\frac{(p_1 \vee p_2 \vee c), (q_1 \vee \neg c)}{p_1 \vee p_2 \vee q_1} \quad (2.2)$$

The necessary condition for the above is that C should not be a function of any of the p_1, p_2, q_1 .

Example 2.5 Show by resolution that $(p \rightarrow q) \rightarrow [(r \wedge p) \rightarrow (r \wedge q)]$ is a tautology:

$$\begin{aligned} &\Rightarrow \neg(\neg p \vee q) \vee [\neg(r \wedge p) \vee (r \wedge q)] \\ &\Rightarrow (p \wedge \neg q) \vee [(\neg r \vee \neg p) \vee (r \wedge q)] \\ &\Rightarrow (p \wedge \neg q) \vee [((\neg r \vee \neg p) \vee r) \wedge ((\neg r \vee \neg p) \vee q)] \\ &\Rightarrow (p \wedge \neg q) \vee [(r \vee \neg r \vee \neg p) \wedge (q \vee \neg r \vee \neg p)] \\ &\Rightarrow (p \wedge \neg q) \vee [(q \vee \neg r \vee \neg p)] \\ &\Rightarrow (q \vee \neg r \vee \neg p \vee p) \wedge (q \vee \neg r \vee \neg p \vee \neg q) \\ &\Rightarrow T \wedge T \\ &\Rightarrow T. \end{aligned}$$

2.6.3 Properties of Inference Rules

An inference rule is a mechanical process of producing new facts from the existing facts and rules. The semantics of predicate logic provides a basis for a formal theory of *logical inference*. It allows the creation of new facts from the existing facts and rules [5, 7].

An interpretation of a predicate statement means the assignment of true or false value to that statement. An interpretation that makes a sentence true is said to *satisfy* a sentence. An interpretation that satisfies every member of a set is said to satisfy the set.

Definition 2.8 (*Logically follows*) If every interpretation that satisfies S also satisfy X , then we say the expression X *logically follows* from a set of expressions S (the *knowledge base*). In other words, the knowledge base S entails sentence X if and only if X is true in all *worlds* where knowledge base is true. If a sentence X logically follows S , we represent it as $S \models X$.

The term logically follows simply means that X is true for every, potentially infinite interpretations that satisfy S . However, it is not a practical way of interpretations. In fact, inference rules provide a computationally feasible way to determine the expression X , when it logically follows a set of premises S .

An example of an inference rule is *Modus Ponens*:

$$[(P \rightarrow Q) \wedge P] \rightarrow Q \quad (2.3)$$

which is a valid statement (a tautology). Here, the Q also *logically follows* (entails) from $(P \rightarrow Q) \wedge P$. That is, $[(P \rightarrow Q) \wedge P] \models Q$.

Definition 2.9 ‘Sound’ inference system.

When every inference X deduced from S also logically follows S , then the inference system is *sound*. This is expressed by,

$$S \vdash x \Rightarrow S \models x. \quad (2.4)$$

The ‘ \vdash ’ is sign of ‘deduction’.

Soundness means that you cannot prove anything that is wrong. \square

Definition 2.10 (A *Complete inference system*) If every X which logically follows S can also be deduced (inferred), then the inference rule is *complete*. This is expressed by

$$S \models x \Rightarrow S \vdash x. \quad (2.5)$$

Completeness means that you can prove anything that is right.

Another rule of inference is *Modus tollens*, specified as,

$$[(P \rightarrow Q) \wedge \neg Q] \rightarrow \neg P \quad (2.6)$$

is sound and complete.

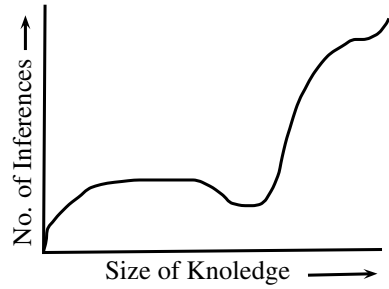
The reader may verify whether the inference rule of modus tollens is sound or complete or both or none?

2.7 Nonmonotonic Reasoning

The classical logic or FOPL (first order predicate logic) discussed far, is not all time sufficient to model the real-world knowledge of the world we live in. The reason are: things become false to true or vice-versa over a time, addition of new knowledge in the knowledge base may contradict the existing knowledge (e.g., the statement “the surface of the earth is curved” becomes false on poles), things may be partially true instead either true or false, and some times there is a probability of being some thing true or false, and so on. Hence, there is a requirement of an all together different approach and method of inferencing for real world situations.

The *Nonmonotonic logic* is the study of those ways of inferring from given knowledge that do not necessarily satisfy the *monotonicity* property, satisfied by all methods

Fig. 2.5 Nonmonotonic reasoning



based on classical logic. In classical logic, if a conclusion is warranted on the basis of certain premises (knowledge), no additional premises will ever invalidate the conclusion.

In everyday life, however, it seems clear that we humans draw sensible conclusions from what we know and that, on the face of new information we often have to take back previous conclusions. This happens even when the new information we gathered in no compel us to take back our previous assumptions (see Fig. 2.5).

For example, we may hold the assumption that “most birds fly”, but that “penguins are birds that do not fly”. On learning that “Tweety is a bird”, we infer that “Tweety flies.” However, on learning that “Tweety is a penguin,” will in no way make us change our mind about the fact that most birds fly, and also that penguins are birds that do not fly or the fact that Tweety is a bird. However, it should make us abandon our conclusion about Tweety’s flying capabilities. It is desirable that intelligent automated systems will have to do the same kind of (nonmonotonic) inferences.

Considering that Γ is a set of sentences of propositional logic, and α is inferred from it, i.e $\Gamma \vdash \alpha$. For any new propositional sentences β , if $\Gamma \cup \{\beta\} \vdash \alpha$ then it is *monotonic* reasoning. If it is not necessary that $\Gamma \cup \{\beta\} \vdash \alpha$, then it is *nonmonotonic* reasoning. We note from Fig. 2.5, that some times, even when we add into knowledge base, the number of inferences decreases instead of increasing; and, this is property of nonmonotonic reasoning.

Some of the systems that perform such nonmonotonic inferences are—*negation as failure, circumscription, modal system, default logic, autoepistemic logic, and inheritance systems.*

2.8 Hilbert and the Axiomatic Approach

An axiomatic system comprises a set of *axioms* and a set of *primitives*, where the primitives are object names but, these objects are left undefined. The axioms are the sentences that make assertions about the primitives. Further, these assertions are not provided with any justifications, so they are neither true nor false. The subsequent or new assertions about the primitives are called *theorems*, are rigorous logical consequences of axioms and previously proved theorems.

In 1899 the mathematician David Hilbert published his ground-breaking research in the form of a book. He provided a complex deductive system based on five *groups of axioms*, namely:

1. Axioms of *incidence*,
2. Axioms of *order*,
3. Axioms of *congruence*,
4. Axioms of *continuity*, and
5. an axiom of *parallels*.

As per Hilbert's approach, the basic concepts of geometry comprises *points*, *lines* and *planes* of Euclidean geometry. However, these concepts are never explicitly defined. Instead, they are implicitly defined by the axioms such that, points, lines, and planes are any family of *mathematical objects* that satisfy the given axioms of geometry.

Twenty years later Hilbert was considered as the chief promoter of a program intended to provide solid foundations to arithmetic, based on purely axiomatic methods—the mathematics that model all the computations. It was called *formalist* program, and Hilbert was identified as the champion of the formalist approach to mathematics as a whole [6].

2.8.1 Roots and Early Stages

The *formal definitions* in an axiomatic system serves the purpose to simplify the things as they can be used to create new objects made of complex combinations of primitives and previously defined terms (objects and theorems). If a definite meaning is assigned to a primitive of an axiomatic system, called as an *interpretation*, the theorems become meaningful assertions.

Following are some definitions of the axiomatic system.

Definition 2.11 (*Model (for axiomatic system.)*) If all the axioms are true for a given interpretation, then everything asserted by the theorem is also true. Such an interpretation is called a model for the axiomatic system.

Definition 2.12 (*Inconsistent (axiomatic system.)*) Since a contradiction can never be true, an axiomatic system using a contradiction can arrive at a logical deduction that it has no model. An axiomatic system with this property is called *inconsistent*.

Definition 2.13 (*Consistent (axiomatic system.)*) If an abstract axiom system does have a model, then such system is consistent.

Definition 2.14 (*Isomorphic*) If two models of the same axiom system can be proved as structurally equivalent, then they are isomorphic to each other.

An axiomatic system can have more than one model.

Definition 2.15 (*Categorical Axioms*) If all models of an axiom system are *isomorphic* then the axiom system is *categorical*.

Thus, for a categorical axiom system, there exists a model—the one and only interpretation in which its theorems are all true.

The qualities—*truth*, *logical necessity*, *consistency*, and *uniqueness* were considered as the base of classical Euclidean geometry. Till recently, it was accepted that Euclidean geometry is the only way to think about space. Now, the axiomatic systems are taken as the basis of geometry, and later all of the mathematics including the computational mathematics and algorithms.

Hilbert's definition of an axiomatic system lays the foundation of *theory* and verifies that this system satisfies three main properties: *independence*, *consistency*, and *completeness*. He proposed that just as in geometry, this kind of axiomatic analysis should be applied to other fields of knowledge, and in particular to physical theories. When we study any system of axioms as per Hilbert's perspectives, the focus of interest remained always on the *disciplines* themselves rather than on the axioms. The axioms are just a means to improve our understanding of the discipline, and not aimed to turn mathematics into a formally axiomatized game. For example, in the case of geometry, a set of axioms were selected in such a way that they reflected the basic manifestations of the intuition of space [4].

2.8.2 Axiomatics and Formalism

To understand the role of axioms, we will discuss the axioms of the set, as they are useful in reasoning and inferences. By analyzing the mathematical arguments, logicians become convinced that the notion of “set” is the most fundamental concept of mathematics. For example, it can be shown that the notion of an integer can be derived from the abstract notion of a set. Thus, in our world all the objects are sets, and we do not postulate the existence of any more primitive objects. To support this intuition, we can think our universe as all sets which can be built by successive collecting processes, starting from the empty set, and we allow the formation of infinite sets.

The first set of axioms for a general set theory was given by E. Zermelo in 1908, and later developed by A. Fraenkel, hence usually referred to as Zermelo-Fraenkel (ZF) set theory, the one we are most concerned. Another systems of axioms, which has only finitely many axioms, but is less natural, was developed by von Neumann, Bernays, and Gödel. The later is usually referred to as Gödel-Bernays (GB) set theory.

Following are some of the important axioms of ZF set theory [3, 8].

1. Axioms of Extensibility.

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y) \quad (2.7)$$

The above says that set is determined by its members. We can define the subsets as follows:

$$x \subseteq y \leftrightarrow \forall z(z \in x \rightarrow z \in y). \quad (2.8)$$

Also,

$$x \subset y \leftrightarrow x \subseteq y \wedge \neg x = y. \quad (2.9)$$

2. *Axiom of the Null set.*

$$\exists x \forall y (\neg y \in x). \quad (2.10)$$

The set defined by this axiom is the empty or null set and we denote it by ϕ .

3. *Axiom of Unordered Pairs.*

$$\forall x \forall y \exists z \forall w (w \in z \leftrightarrow w = x \vee w = y). \quad (2.11)$$

We represent the set z by $\{x, y\}$. Also, $\{x\}$ is $\{x, x\}$ and we put $\langle x, y \rangle = \{\{x\}, \{x, y\}\}$. The set $\langle x, y \rangle$ is called *ordered pair* of x and y .

Using the above we can define a function as follows: a function is a set f of ordered pairs such that $\langle x, y \rangle, \langle x, z \rangle \in f \rightarrow y = z$. The set of x such that $\langle x, y \rangle \in f$ is called *domain*, and set of y is called *range*. We say, f maps in set u if the range of f is in u .

4. *Axiom of set Union.* It can be expressed as:

$$\forall x \exists y \forall z (z \in y \leftrightarrow \exists t (z \in t \wedge t \in x)). \quad (2.12)$$

The above says that y is union of all sets in x . Using the axiom Eq. 2.12, we can deduce that given x and y , there exists z , such that $z = x \cup y$, that is, $t \in z \leftrightarrow t \in x \vee t \in y$.

To motivate for the next axiom being described, if x is an integer, the successor of x will be defined as $x \cup \{x\}$. Then the ‘‘axiom of infinity’’ generates a set that contains all the integers and thus infinite.

5. *Axiom of Infinity.* It can be expressed as follows, and we understand that it is the principle of Induction.

$$\exists x (\phi \in x \wedge \forall (y \in x \rightarrow y \cup \{y\} \in x)). \quad (2.13)$$

6. *Axiom of Power Set.* This axiom states that there exists for each x the set y for all the subsets of x .

$$\forall x \exists y \forall z (z \in y \leftrightarrow z \subseteq x). \quad (2.14)$$

If the axiom of extensionality is dropped, the resulting system may contain atoms, i.e., sets x such that $\forall y (\neg y \in x)$ yet the sets x are different. Indeed, one possible view is that integers are atoms and should not be taken as sets.

The first interesting axiom is the Axiom of Infinity. If we drop it, then we can take a model for ZF set of all finite sets which can be built from ϕ .

The axioms discussed above can be used to prove theorems, like, *mathematical induction*, *invertible functions*, and in fact another theorem of set theory, as well as the corollaries, but the same are not appropriate to cover here, and a curious reader is encouraged to refer the literature given in the bibliography.

2.9 Summary

Logic is used for valid deductions, and it avoids fallacy reasoning. Logic is also useful in *argumentation theory*—a study of how conclusions can be reached through logical reasoning, that is, whether the claims are soundly based on premises or not. Argumentation includes debate and negotiation, that are concerned with reaching mutually acceptable conclusions. The logic is used in proofs, games, and puzzles solutions. The arguments have the internal structure: comprising of premises, reasoning process, and consequence.

The most commonly used, Propositional logic, represents sentences using single symbols, called *atoms*, which are joined using the operators \vee , \wedge , \neg , \rightarrow to create compound sentences. The sign of “ \rightarrow ” in $p \rightarrow q$ is *material implication*, also called *conditional join*, if p then q . Propositional logic expressions are called sentences/statements; these are interpreted as *true* or *false*. The sentences are called *wff*, and are defined recursively. A formula is a *syntactic* concept, which means whether or not a string of symbols is a formula.

The meaning (*semantics*) is associated with each formula by defining its *interpretation*, which assign a value *true* (T) or *false* (F) to every formula. Interpretation of a statement means the assignment of true values to its atoms. A set of truth values assigned to the atoms in a statement is called its *world*. Assignment of truth values to the atoms in a statement, which makes the statement true is called *model* of the statement.

The model checking is the process of truth-table enumeration, and is exponential on n , the number of atoms in a statement. The derivation can also be represented by a *derivation-tree* (*parse-tree*).

A propositional formula A is *satisfied* iff $v(A) = \text{True}$ for some interpretation v . A satisfying interpretation is called *model* for A . The formula A is called *valid*, denoted by $\models A$, iff $v(A) = \text{True}$ for all interpretations v . A sentence is logically true (valid) iff it is true under every interpretation. $\models \theta$ means that θ is valid.

A reasoning, in which addition of new knowledge may produce inconsistency in the knowledge base, is called *nonmonotonic reasoning*. As per the property of *monotonicity*, if $S \vdash \alpha$, and β is additional assertion, then $S \wedge \beta \vdash \alpha$. The *Nonmonotonic logic* is the study of those systems that do not satisfy the *monotonicity* property satisfied by all methods based on classical logic.

The reasoning pattern comprises *inference methods*: *modus ponens*, *modus tollens*, *syllogism*; and *Proof methods*: *resolution theorem*, *model checking*, *model checking*,

Normal forms. Deducing new knowledge from the existing set of the knowledge base is called *inferencing*. The *Modus ponens*, *modus tolens*, *sylllogism* are inference rules, and *Sound* and *Complete* are good properties of inference systems.

Semantic tableau is a method for deciding satisfiability for the formula of propositional calculus, which *systematically* searches for a model for a formula. If it is found the formula is satisfiable, else not satisfiable. *Semantic Tableau* is a tree, each node of which will be labeled with a set of formulas, and these formulas are inductively expanded to leaves such that each leaf is marked as *open* by \odot or *closed* by \times .

The *resolution rule* is an inference which uses *deduction* approach. It is used in theorem proving.

If every interpretation that satisfies S also satisfy X , then we say expression X *logically follows* from a set of expressions S (the *knowledge base*). The *Soundness* means that you cannot prove anything that is wrong, and *Completeness* means that you can prove anything that is right.

An axiomatic system comprises a set of *axioms* and *primitives*.

Exercises

1. Prove the following assertions:
 - a. α is valid only if $\mathbf{T} \models \alpha$, where \mathbf{T} stands for True.
 - b. For any (not necessarily valid) α , $\mathbf{F} \models \alpha$, where \mathbf{F} is logically False.
 - c. $\Gamma \models \beta$ if and only if the sentence $(\Gamma \Rightarrow \beta)$ is valid. Here, *Gamma* is knowledge base and β is sentence.
 - d. $\alpha \equiv \beta$ if and only if the sentence $(\alpha \Leftrightarrow \beta)$ is valid. Here, α and β are sentences.
2. Give your argument in favor and against, that the *material join* $p \rightarrow q$ is not same as the *cause effect* $p \rightarrow q$.
3. Determine, which of the following formulas are valid /satisfied /contradiction?
 - a. $((p \rightarrow q) \wedge (\neg p \rightarrow r)) \rightarrow (q \vee r)$
 - b. $(p \vee q) \rightarrow (p \wedge q)$
 - c. $p \rightarrow \neg q$
 - d. $(p \wedge q) \rightarrow (p \vee q)$.
4. Show that $(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$ is valid (Hint: Construct truth-table, the interpretation of this formula shall be true for all the worlds).
5. Show that formula $(\neg p \wedge \neg q) \wedge (p \vee q)$ is unsatisfiable (Hint: Construct truth-table).
6. Assume a vocabulary with only four propositions, A , B , C , and D . Find out the number of *models* for each of the following sentences?
 - a. $P \Leftrightarrow Q \Leftrightarrow R$.
 - b. $P \wedge Q$,
 - c. $(P \wedge Q) \vee (Q \wedge R)$,

7. If S is a set of propositional formulas, then show that $S \models F$ iff $S \cup \{\neg F\}$ is contradiction (Hint: A set of propositional formulas is contradiction, iff every valuation of S there is a formula p in the set such that $S \not\models p$).
8. If $\theta \models \psi$ then prove that $\theta \cup \{A\} \models \psi$ for any formula A .
9. If $\theta \models \psi$ and A is valid then prove that $\theta - \{A\} \models \psi$ for any formula A .
10. Find out the satisfiability of the following formulas using semantic Tableau methods:
 - a. $p \wedge (\neg p \vee \neg q)$.
 - b. $(p \wedge q) \vee (\neg p \wedge \neg q)$.
11. Show that if S is unsatisfiable then $S - \{A_i\}$ is also unsatisfiable for every $i \leq n$.
12. Establish the consistency/inconsistency of the following statements using Tableau method.
 - a. $(p \wedge q) \vee (p \rightarrow q)$,
 - b. $(\neg q \rightarrow \neg p) \leftrightarrow (p \rightarrow q)$,
 - c. $(\neg p \vee \neg q) \wedge (p \vee q)$.
13. Convert the following statements into CNF:
 - a. $a\neg bc + bc + \neg ab\neg c$,
 - b. $a + bc + \neg b\neg c$.
14. Convert the following statements into DNF:
 - a. $(a + b)(a + \neg b + c)(a + \neg c)$,
 - b. $(\neg a + b) \vee (a \rightarrow b \rightarrow c)$.
15. Find out the resolvent for $\{p \rightarrow q, \neg q \vee \neg r, r \rightarrow p\}$.
16. Write a recursive algorithm $TRUE(\mathbf{x}, \mathbf{M})$ that returns true if and only if propositional logic sentence x is true in the model \mathbf{M} , where \mathbf{M} assigns a truth value for every symbol in x . The algorithm should run in time linear in the size of the sentence.

References

1. Ben-Ari M (2008) Mathematical logic for computer science, 2nd edn. Springer International, London. ISBN-978-81-8128-344-3
2. Chowdhary KR (2012) Fundamentals of discrete mathematical structures, 2nd edn. EEE PHI India. ISBN: 978-81-203-4506-5
3. Cohen PJ (2008) Set theory and the continuum hypothesis. Dover, New York
4. Davis M, Putnam H (1960) A computing procedure for quantification theory. J ACM 7(3):201–215. <https://doi.org/10.1145/321033.321034>
5. Gelder AV (1991) The well-founded semantics for general logic programs. J ACM 38(3):620–650
6. Hilbert D (2005) The foundations of geometry. Ebook # 17384

7. Kaushik S (2002) Logic and prolog programming. New Age International, New Delhi
8. Pappas P (1972) Axiomatic set theory. Dover, New York
9. Shankar N (2009) Automated deduction for verification. ACM Comput Surv 41(4):20:1. <https://doi.org/10.1145/1592434.1592437>