The Completeness Theorem of Gödel

1. An Introduction to Mathematical Logic

S M Srivastava

This is a two-part article giving a brief introduction to mathematical logic. It will culminate in the so-called completeness theorem of Kurt Gödel, which will be proved in the second part.

1. Introduction

We are all aware of the fundamental difference between methods of the mathematician and those of the natural scientist. A natural scientist establishes laws of nature mainly through observation and experiment while a mathematician must produce 'proof' to establish laws of mathematics. For instance, a mathematician may measure the angles of many triangles and conclude that the sum of the angles is 180°. However, it will not be accepted as a law of mathematics until it is proved.

S M Srivastava is with the Indian Statistical Institute, Calcutta. He received his PhD from the Indian Statistical Institute in 1980. His research interests are in descriptive set theory.

So, what is the method of mathematicians? This highly philosophical question was seriously addressed by some of the leading mathematicians including Russell, Hilbert and von Neumann around the beginning of the twentieth century. This was in response to some paradoxes and some of the hardest problems (such as the continuum hypothesis) that appeared in mathematics following the creation of set theory by Georg Cantor. Their work laid the foundation of a very beautiful branch of mathematics, called *Mathematical Logic.* Finally, in what is lauded as epochal work constituting an achievement of the first order, Gödel developed powerful techniques to prove very deep and stunning results such as completeness theorem for first order logic, incompleteness theorem, consistency of the continuum hypothesis and so on. This is an introductory article on this branch of

A natural scientist establishes laws of nature mainly through observation and experiment while a mathematician must produce 'proof' to establish laws of mathematics.

There has to be first laws, called *axioms*, which we accept without proof.

mathematics culminating in the completeness theorem for first order logic.

2. Languages

Obviously, not all laws of mathematics can be proved. There has to be first laws, called *axioms*, which we accept without proof. These axioms should be simple and clear statements which, for some reason or other, are considered evident. All other laws, called *theorems*, must be deduced from axioms following certain rules of inference. Just as there has to be certain first laws, there has to be certain first concepts which are left undefined. Other concepts are defined in terms of the undefined ones. To understand what we mean, let us recall the well-known case of geometry as developed by the Greek mathematicians like Euclid. Starting from a few undefined concepts like '*points*', '*lines*', 'a point *incident* on a line', 'a point lying *between* two points of a line', '*congruent* triangles', etc. and a set of five axioms, they deduced many laws (i.e., theorems) of plane geometry.

Usually, in mathematics, axioms are stated in a natural language such as English or Hindi or Bengali. However, it is desirable to use a *formal or artificial language* with precise rules of formation for linguistic objects like names for individual objects (i.e., nouns), sentences, etc. Or, so to speak, the language should have a precise *grammar*. There are two main reasons for this. The first reason is to avoid ambiguities. For example, set theory has an axiom asserting that 'given a set A and a property P, there is a set consisting of all elements of A having the property P'. But what is a property?

All other laws, called *theorems*, must be deduced from axioms following certain rules of inference.

The second reason for using a formal language is quite deep. Once we have fixed our axioms, it is quite conceivable that for a statement P, neither P nor its negation is provable from the axioms using legitimate rules of inference. In such a case we say that P is *independent* of the axioms. For example, in the nineteenth century, the parallel axiom of Euclid was shown to be independent of the remaining axioms and one of the most remarkable discoveries of twentieth century mathematics is the independence of the continuum hypothesis as shown by Gödel and Cohen. Clearly, to prove such independence results, we must make the intuitive notion of proof or logical deduction precise. This is best done by using an artificial language.

In the nineteenth century, the parallel axiom of Euclid was shown to be independent of the remaining axioms and one of the most remarkable discoveries of twentieth century mathematics is the independence of the continuum hypothesis as shown by Gödel and Cohen.

We now proceed to define a class of artificial languages, known as *first order languages*, which have sufficiently high expressive power. Such languages contain 'variables' x, y, etc., for generic representation of objects or individuals of our study. It also contains symbols like $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\exists$, $\forall$ to intuitively mean 'not', 'or', 'and', 'implies', 'for some', 'for all', respectively. These symbols, together with $=$ (to mean 'equality'), will be called *logical symbols*. They are present in all first order languages. Moreover, a first order language may contain some *non-logical symbols* representing undefined concepts of the underlying theory. For example, if we are studying the theory of natural numbers, we write

$$\neg \exists x(x < 0) \qquad (1)$$

to express the fact that 'zero is the smallest element'. Here $<$ and 0 are non-logical symbols. Similarly, if we are studying the theory of real numbers, the string of symbols

$$\forall x(x = 0 \vee \exists y(x \quad y = 1 \wedge y \quad x = 1)) \qquad (2)$$

expresses that 'every non-zero real number has a reciprocal'. Here 0, 1 and   are non-logical symbols.

More precisely, a *first order language* L consists of

(a) **variables**: a countably infinite set of symbols, generically denoted by x, y, z, with or without suffixes.

(b) **logical connectives**: $\neg$, $\vee$

The language for group theory has one constant symbol 1 and one 2-ary or binary function symbol ... and it has no other non-logical symbols.

(c) **existential quantifier**: $\exists$

(d) **equality symbol**: $=$

(e) **constant symbols**: a (empty or non-empty) set of symbols

(f) **function symbols**: for each $n \geq 1$, a (empty or non-empty) set of symbols.

(g) **relation symbols**: for each $n \geq 1$, a (empty or non-empty) set of symbols.

Symbols occurring in (a)-(d) are common to all first order languages. These are called *logical symbols*. Symbols occurring in (e)-(g), known as *non-logical symbols*, depend on the particular theory for which the language is defined. For example, the language for group theory has one constant symbol 1 and one 2-ary or binary function symbol and it has no other non-logical symbols; the non-logical symbols for the language for an ordered field are: two constant symbols 0 and 1, two binary function symbols $+$ and and a binary relation symbol $<$. It is assumed that the logical and non-logical symbols are all distinct.

In the next few paragraphs, L denotes a first order language. We give precise rules of formation of linguistic objects, represented by finite strings of symbols, of the language L. There are two major classes of such objects. Somewhat broadly speaking, the first kind, called *terms*, can be thought of as names for the individual objects of study, i.e., the nouns and the second kind, called *formulae*, can be thought of as statements made about these individuals.

We define the *terms* of L by induction as follows:

(a) Each variable is a term.

(b) Each constant symbol is a term.

(c) If t_1, t_2, ..., t_n are terms and f a n-ary function symbol, then $ft_1t_2\ldots t_n$ is a term.

A string of symbols of the form

(i) $t = s$, where t, s are terms, or

(ii) $pt_1t_2\ldots t_n$, where p is a n-ary relation symbol of L and t_1, t_2, ..., t_n are terms

is called an *atomic formula* of L.

We define the *formulae* of L by induction as follows:

(a) Each atomic formula is a formula.

(b) If A is a formula, so is $\neg A$.

(c) If A and B are formulae, so is $A \vee B$.

(d) If A is a formula and x a variable, then $\exists x A$ is a formula.

We shall use parantheses in a natural way to avoid ambiguities. For instance, we shall write $\neg(P \vee Q)$, and not $\neg P \vee Q$, if we mean so. On the other hand, we shall adopt the convention of association to the right for omitting parantheses. So, $A \vee B \vee C$ is really the formula $A \vee (B \vee C)$ and $A \vee B \vee C \vee D$ is the formula $A \vee (B \vee (C \vee D))$.

Is the expressive power of a first order language sufficiently high? For instance, in (2) we used, among others, the logical symbols $\forall$ and $\wedge$ to express that every non-zero real number has a reciprocal. But these are not symbols of a first order language. We have not introduced these in our language for reasons of economy. However, they can be defined in terms of $\neg$ and

We choose the language and its grammar suitably so that the syntactical structure of the sentence reflects its meaning.

$\vee$ as follows: $A \rightarrow B$ will stand for $\neg A \vee B$; $A \wedge B$ for $\neg(\neg A \vee \neg B)$; $A \leftrightarrow B$ for $(A \rightarrow B) \wedge (B \rightarrow A)$ and $\forall x A$ for $\neg \exists x \neg A$. According to our convention of association to the right, $A \rightarrow B \rightarrow C$ is the formula (in abbreviated form) $A \rightarrow (B \rightarrow C)$.

We have already gained a bit by expressing mathematical statements as a string of symbols. To see this we note that mathematical concepts are abstract and so, difficult to understand. On the other hand, a formula, being a finite string of symbols, is a concrete object. Thus we have concrete objects representing a mathematical statement. This enables us to approach abstract through concrete. We make use of this as follows. We choose the language and its grammar suitably so that the syntactical structure of the sentence reflects its meaning. Then from any set of sentences we can deduce others following certain mechanical rules of inference. This view, known as the *formalist's* view, was advanced by the famous German mathematician David Hilbert.

We close this section by giving a few technical definitions.

For any formula A, we define the set of all *subformulae* of A by induction on the length of A as follows:

(i) If A is an atomic formula, then A is the only subformula of A.

(ii) If A is the formula $\neg B$ or the formula $\exists x B$, then A is a subformula of A and every subformula of B is a subformula of A.

(iii) If A is the formula $B \vee C$, then A is a subformula of A and every subformula of B or C is a subformula of A.

We think of subformulae of A as building blocks of the formula A. An occurrence of a variable x in a formula

A is called *bound* if it occurs in a subformula of A of the form $\exists xB$; otherwise it is called a *free* occurrence. A variable x is called *free* in A if it has a free occurrence in A. A formula A is called *closed* or a *sentence* if no variable is free in A. In the sequel, we shall write $A[x_1, \cdots, x_n]$ to indicate that no variable other than $x_1, \cdots, x_n$ is free in A. Similarly, we shall write $t[x_1, \cdots, x_n]$ to indicate that no variable other than $x_1, \cdots, x_n$ occur in the term t.

If a, t are terms and x a variable, then $t_x[a]$ denotes the term obtained from t by simultaneouly replacing each occurrence of x in t by a. Certain caution has to be exercised for substituting a term in a formula. For instance, suppose A is the formula of number theory $\exists y(x = 2 \cdot y)$ meaning that x is even. Now, if we substitute $y + 1$ for x in A the meaning of the formula will completely change. For such reasons, we say that a term t is *substitutable* for a variable x in a formula A, if for each variable y occurring t, no subformula of A of the form $\exists yB$ contains an occurrence of x that is free in A. If t is substitutable for x in A, then the formula obtained from A by simultaneously replacing each free occurrence of x in A by t is denoted by $A_x[t]$. Also, suppose a term t_i is substitutable for the variable x_i in the formula $A[x_1, \cdots, x_n]$, $1 \leq i \leq n$. Then $A_{x_1,\ldots,x_n}[t_1, \cdots, t_n]$ denotes the formula obtained from A by simultaneously replacing each free occurrence of x_i in A by t_i, $1 \leq i \leq n$. Similarly, for any term $a[x_1, \cdots, x_n]$, $a_{x_1,\ldots,x_n}[t_1, \cdots, t_n]$ is defined by simultaneously replacing each occurrence of x_i in a by t_i, $1 \leq i \leq n$.

In propositional logic there is no object or individual of discourse. Instead, starting from some sentences, called *atomic formulae*, we form formulae using only the logical connectives $\neg$ and $\vee$. The main aim is to formalize the logic or the rules of inference involving logical connectives only.

For simplicity, we shall divide our study in two parts: *propositional logic* and *first order logic*. In propositional logic there is no object or individual of discourse. Instead, starting from some sentences, called *atomic formulae*, we form formulae using only the logical connectives $\neg$ and $\vee$. The main aim is to formalize the logic or the rules of inference involving logical connectives only.

In the syntactical approach a formula is regarded just as a string of symbols and logical deductions are made depending only on the structure of formulae.

Thus *language for a propositional logic* L consists of

(i) **atomic formulae**: a non-empty set of symbols, and

(ii) **logical connectives**: $\neg$ and $\vee$.

If L is the language for a propositional logic, the *formulae* of L are defined by induction as follows:

(i) Each atomic formula is a formula of L,

(ii) If A and B are formulae of L, so are $\neg A$ and $A \vee B$.

Propositional logic (as well as several other logic including first order logic) is of great interest in computer science. But we shall limit its study only to facilitate our study of first order logic.

3. What is a Proof?

We now turn our attention to the fundamental notion of a *logical deduction* or a *proof*. There are two approaches: semantic (also known as logicism) and syntactical (also known as formalism). In the semantic approach one takes into account all possible meanings of atomic formulae. In the syntactical approach a formula is regarded just as a string of symbols and logical deductions are made depending only on the structure of formulae.

We explain the semantic approach first. Since our main goal is to define the notion of a logical deduction, it is natural to start with the knowledge of which of the atomic formula is true and which is false. In the next step we use the intuitive meaning of the logical connectives and quantifiers and define the truth or falsity of a sentence in terms of its subformulae.

3a. Interpretation or Semantics of Propositional Logic

Propositional logic is of great interest in computer science.

From now on, in this section, L denotes a language for a propositional logic and by a formula we mean a formula of L.

Definition. *A truth valuation* or an *interpretation* or a *structure* of L is a map v from the set of all atomic formulae of L to $\{T, F\}$.

Definition. Let v be an interpretation of L. We extend v (and denote the extension by v itself) to the set of all formulae by induction as follows:

$$v(\neg A) = T \text{ if and only if } v(A) = F$$

and

$$v(A \vee B) = T \text{ if and only if } v(A) = T \text{ or } v(B) = T$$

If $v(A) = T$ we say that A is *true* in the structure v.

Exercise 3.1. Let A, B be formulae and v a truth valuation of L. Show the following.

(a) $v(A \wedge B) = T$ if and only if $v(A) = v(B) = T$

(b) $v(A \rightarrow B) = T$ if and only if $v(A) = F$ or $v(B) = T$

(c) $v(A \leftrightarrow B) = T$ if and only if $v(A) = v(B)$.

Definition. Let $\mathcal{A}$ be a set of formulae. An interpretation v is called a *model* of $\mathcal{A}$ if every $A \in \mathcal{A}$ is true in v. In this case we write $v \models \mathcal{A}$.

Definition. Let A, B be formulae and $\mathcal{A}$ a set of formulae.

(i) We say that A is a *tautological consequence* of $\mathcal{A}$, and write $\mathcal{A} \models A$, if A is true in every model v of $\mathcal{A}$.

(ii) If A is a tautological consequence of the empty set of formulae, we say that A is a *tautology* and write $\models A$. Thus, A is a tautology if and only if $v(A) = T$ for every truth valuation v of L.

(iii) If $A \leftrightarrow B$ is a tautology (i.e., if $v(A) = v(B)$ for all truth valuation v), we say that A and B are *tautologically equivalent.*

Exercise 3.2.

(a) Show that $A \to B \to C$ and $B \to A \to C$ are tautologically equivalent.

(b) Show that $A \to B \to C$ and $A \to C \to B$ are not tautologically equivalent.

(c) Show that $A \to B \to C$ and $(A \to B) \to C$ are not tautologically equivalent.

Exercise 3.3. Let $A, A_1, A_2, \ldots, A_n$ be formulae. Show that the following statements are equivalent:

(a) A is a tautological consequence of $A_1, A_2, \ldots, A_n$.

(b) $A_1 \to A_2 \to \cdots \to A_n \to A$ is a tautology.

3b. Semantics of First Order Languages

Throughout this section, unless otherwise stated, L denotes a first order language. Further, by terms, formulae, etc. we mean terms, formulae, etc. of L.

A *structure* or an *interpretation* $\mathcal{M}$ of L consists of the following:

(i) a non-empty set M, called the *universe* of $\mathcal{M}$; (Elements of M are called *individuals.*)

(ii) for each constant symbol c of L, an element $\mathcal{M}(c)$ of M;

(iii) for each n-ary function symbol f of L, a function $\mathcal{M}(f)$ from M^n to M; and

(iv) for each n-ary relation symbol p of L, a n-ary relation $\mathcal{M}(p)$ on M

$\mathcal{M}(c)$, $\mathcal{M}(f)$ and $\mathcal{M}(p)$ are called the *interpretations* or *meanings* of c, f and p, respectively in $\mathcal{M}$. The meaning of a variable-free term t of L in a structure $\mathcal{M}$ is easy to define now. Indeed, we already have the meaning $\mathcal{M}(c)$ of any constant symbol c. We extend this definition to all variable-free terms by induction on the length of t: we define $\mathcal{M}(ft_1 \;\; . t_n)$ to be $\mathcal{M}(f)(\mathcal{M}(t_1) \;\; . \mathcal{M}(t_n))$, where $t_1, \;\; \cdot, t_n$ are variable-free terms and f a n-ary function symbol.

We now define truth or falsity of a formula A in a structure $\mathcal{M}$. We need to extend L a bit so as to have a constant symbol corresponding to each individual m. (The reason for doing this will become clear a little later.) Given L and $\mathcal{M}$, $L(\mathcal{M})$ denotes the first order language obtained from L by adding a new constant symbol i_m for each $m \in M$ Clearly each formula of L remains a formula of $L(\mathcal{M})$. We shall regard $\mathcal{M}$ as a structure for $L(\mathcal{M})$ by defining $\mathcal{M}(i_m) = m$, meanings of the non-logical symbols of L remaining the same.

Let t and s be variable free terms of $L(\mathcal{M})$. We say that the formula $t = s$ is true in $\mathcal{M}$ if $\mathcal{M}(t) = \mathcal{M}(s)$. Similarly, we say that a variable free atomic formula $pt_1 \;\; . t_n$ of $L(\mathcal{M})$ is true in $\mathcal{M}$ if $\mathcal{M}(p)(\mathcal{M}(t_1), \;\;\; , \mathcal{M}(t_n))$ holds. Inductively, we extend this notion for all closed formulae of $L(\mathcal{M})$. We say that $\neg A$ is true in $\mathcal{M}$ if A is false, i.e., A is not true in $\mathcal{M}$; $A \vee B$ is true in $\mathcal{M}$ if at least one of A and B is true in $\mathcal{M}$; we say that $\exists x A$ is true in $\mathcal{M}$ if there is an individual $m \in M$ such that the closed

The formulae of a first order language L are precisely the formulae of the language of the propositional logic whose atomic formulae are the elementary formulae of L.

formula $A_x[i_m]$ is true in $\mathcal{M}$. Finally, $A[x_1, \ldots x_n]$ is defined to be true in $\mathcal{M}$, if for every $m_1, \ldots, m_n$ in M the sentence $A_{x_1,\ldots,x_n}[i_{m_1}, \ldots, i_{m_n}]$ is true in $\mathcal{M}$. Note that to define the truth or falsity of closed formulae of the from $\exists x A$ or formulae which are not closed, we needed to introduce the language $L(\mathcal{M})$.

Exercise 3.4. Let L be a first order language, $\mathcal{M}$ a structure for L and $t_1, \ldots, t_n$ variable-free terms of L. Assume that $\mathcal{M}(t_i) = m_i$, $1 \leq i \leq n$. Show the following:

[a] For any term $a[x_1, \ldots, x_n]$,

$$\mathcal{M}(a_{x_1,\ldots,x_n}[t_1, \ldots, t_n]) = \mathcal{M}(a_{x_1,\ldots,x_n}[i_{m_1}, \ldots, i_{m_n}]).$$

[b] For any formula $A[x_1, \ldots, x_n]$, $A_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$ is true in $\mathcal{M}$ if and only if $A_{x_1,\ldots,x_n}[i_{m_1}, \ldots, i_{m_n}]$ is so.

We call $\mathcal{M}$ a *model* of a set of formulae $\mathcal{A}$ of L, and write $\mathcal{M} \models \mathcal{A}$, if every formula A belonging to $\mathcal{A}$ is true in $\mathcal{M}$. If A is a formula of L and $\mathcal{A}$ a set of formulae, we say that A is a *logical consequence* of $\mathcal{A}$, and write $\mathcal{A} \models A$, if A is true in every model of $\mathcal{A}$.

Some formulae are true in a structure because of particular properties of the structure. On the other hand, some formulae are true in all structures simply because of the meaning of the logical symbols.

The next few definitions will be needed for adapting some results from propositional logic to first order logic. We call a formula *elementary* if either it is an atomic formula or a formula of the form $\exists x B$. It is easy to see that the formulae of L are precisely the formulae of the language of the propositional logic whose atomic formulae are the elementary formulae of L. A *truth valuation* for L is a map v from the set of all elementary formulae into $\{T, F\}$. We can extend v as before to the set of all formulae. Further, we define the notion of *tautological consequences*, *tautology* and *tautologically equivalent formulae* in exactly the same way as before.

In the second part of this article, we shall start with the syntactical definition of a proof. This is based on two very important observations. We know that some formulae are true in a structure because of particular properties of the structure. On the other hand, some formulae, e.g., those of the form $\neg A \vee A$ or $x = x$ or $A_x[t] \rightarrow \exists x A$, are true in all structures simply because of the meaning of the logical symbols. Similarly sometimes a formula is inferred from a set of formulae because of the meaning of logical symbols. For example, the formula B is true in all those structures in which A and $A \rightarrow B$ are true and $\exists x A \rightarrow B$ is true in all those structures in which $A \rightarrow B$ is true provided x is not free in B. To define a proof syntactically, we shall have to fix some formulae which are true in all structures and call them *logical axioms.* Further, we shall fix some *rules of inference.* This will be the starting point for the second part.

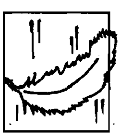
To define a proof syntactically, we shall have to fix some formulae which are true in all structures.

I would like to thank B V Rao, Amartya Dutta and H Sarbadhikari for their suggestions on improvement.

Suggested Reading

[1] Joseph R Shoenfield, *Mathematical Logic*, Addison-Wesley, 1967.

Address for Correspondence
S M Srivastava
Stat-Math Unit
Indian Statistical Institute
203 B T Road
Calcutta 700 035, India.
e-mail: smohan@isical.ernet.in



Collecting fresh fruits becomes even harder as the tree of knowledge grows higher and wider. However, there are certain branches that provide surer footholds to the new growths, and teachers must search these out.

J E Caroll
Rate equations in semiconductor electronics
Cambridge University Press, Cambridge
1985, p.vi.