

Mathematical Limits

Logic and Computational Theorems

Amir H. Ebrahimnezhad

June 6, 2023

Contents

1	Ground Works	1
1.1	Vorwort	1
2	Formal Language Theory	1
2.1	Languages	1

1 Ground Works

1.1 Vorwort

Gödel's incompleteness theorems, are two theorems of mathematical logic that deal with the limits of provability in formal axiomatic theories. The theorems are interpreted as showing that Hilber's program to find a complete and consistent set of axioms for all mathematics is impossible.

In the following work we will first describe the theorem and preliminaries, and then prove it using algorithm theory methods and other methods. Then we are going to talk about it's consequences in pure mathematics and philosophy of mathematics, later we will discuss the relation between the theorems and the church-turing thesis We then would discuss related works that has pushed the theorem further and shown aspects of it. At the final section we will investigate a philosophical picture of Mathematical Universe and if it is acceptable to consider a bigger abstract world of mathematical objects where the real world is a subset of it.

2 Formal Language Theory

The investigation of Incompleteness theorem requiers the knowledge of some basic ideas such as languages, grammars, automatas and others. Therefore we start this draft by talking about these topics in a general format. A more detailed version would be published later.

2.1 Languages

We would begin by the definition of a language. In a informal way a language is what we speak or what we write on a paper to give information to another person, mathematically and a more abstract idea of a language is a bunch of symbols that we have to make words (gluing symbols together to resemble a unity of them); To give a precise mathematical definition of language we would first define alphabet as:

Definition 1 –An alphabet Σ , is a set of symbols.

Definition 2 –A word M , is a combination of symbols from alphabet Σ .

For instance one could choose the set $\{a, b, c, d, \dots\}$ as ones alphabet. Then "cat" is a word in it's alphabet. As you can see there's a meaning for the word we gave as an example. It defines an object (more accurately a living creature). Since there can be Infinitely many words for any alphabet (other than an empty set which have no words), we distinguish between the accepted combinations and the ones we don't accept by defining a grammar for our language; A grammar is a way to characterize a language, a way to list which strings of Σ is acceptable. We could simply list strings or have a set of rules (or an algorithm) to say if a given combination is acceptable or not for a language. Thus we define a language as:

Definition 3 –Given an alphabet Σ , Σ^∞ is the set of all possible words in the alphabet.

Definition 4 –A subset S of a set X is decidable if and only if there exists a function that given $x \in X$ decides if $x \in S$ is true or false.

Definition 5 –A Language L , is a subset of the alphabet Σ^∞ ($L \subset \Sigma^\infty$) where there exists a function $\eta(\sigma \in \Sigma^\infty)$ called grammar that decides L .

Formally, we define a grammar as:

Definition 6 –A Grammar is a set $\{V_T, V_N, S, R\}$ where V_T is the set of terminal elements, V_N is the set of non-terminal elements, S is a memeber of V_N , and R is a finite set of rules.

We would use these definitions in the later sections of this draft. But for now let us give a formal definition of R as well:

Definition 7 – R is a finite set of ordered pairs from $\Sigma^\infty V_N \Sigma^\infty \times \Sigma^\infty$, where $\Sigma = V_T \cup V_N$.

In later drafts we would get back to the Formal language theory in more depth and examine the properties of the setof languages each grammar formalism can accomodate and the set of abstract machines that correspond to each type. But for now we would start with the things we need for proving the incompleteness theorems that is the goal of this draft.

We assume that we are given a subset T of language L , which is called the set of *true statements*. This set should contain only the statements in the language that would evaluate as True. In the process of assuming such subset we omit the part where we would consider the statements true or false. A language then would be completely defined (for our purpose) if we are given the *fundamental pair*:

Definition 8 –Given a language L and the subset of true statements T , we call $\langle L, T \rangle$ a *fundamental pair*.

► **Induction:** Induction is a method of proving that a statement $P(n)$ is true for every natural number n , that is, that the infinitely many cases, $P(0), P(1), \dots$ all hold. [1]

"Mathematical induction proves that we can climb as high as we like on a ladder, by proving that we can climb onto the bottom rung (the basis) and that from each rung we can climb up to the next one (the step)."

Theorem 1 — For every natural number n ,

$$1 + 2 + \dots + n = \frac{n(n+1)}{2} \quad (1)$$

proof: If $n = 1$, the equality holds. For the inductive case, fix $k \geq 1$ and assume that:

$$1 + 2 + \dots + k = \frac{k(k+1)}{2} \quad (2)$$

Now adding $k + 1$ to each side we have:

$$1 + 2 + \dots + (k+1) = \frac{k(k+1)}{2} + (k+1) \quad (3)$$

Since the right hand side simplifies to:

$$\frac{(k+1)((k+1)+1)}{2} \quad (4)$$

Finishing the inductive step and thus the proof. As you can see in the inductive step what we prove is that:

"If the formula holds for k , then the formula holds for $k + 1$."

Looking at this from a slightly different angle, what we have done is to construct a set of numbers with a certain property. If we let S stand for the set of numbers for which our theorem holds, in our proof by induction we show that the set S , is identical with the set of natural numbers, thus the theorem holds for every natural number n , as needed.

So what makes a proof by induction work is the fact that the natural numbers can be defined recursively, There is a base case, consisting of the smallest natural number, and there is a recursive case, showing how to construct bigger natural number from smaller ones.

► **Terms and Formulas:** As we mentioned earlier not all words of a set Σ^∞ is meaningful. Since any combination of the alphabet is a word there has to be distinctions between what are meaningful words and what are not. We would consider two kinds of words as *terms* & *formulas* as follow:

Definition 9 –If \mathcal{L} is a language, a **term** of \mathcal{L} is a nonempty finite string t of symbols from \mathcal{L} such that either:

1. t is a variable, or
2. t is a constant symbol, or
3. $t \equiv f t_1 t_2 t_3 \dots t_n$, where f is an n -ary function symbol of \mathcal{L} and each of the t_i is a term of \mathcal{L} .

Definition 10 –If \mathcal{L} is a language, a **formula** of \mathcal{L} is a nonempty finite string ϕ of symbols from \mathcal{L} such that either:

1. $\phi \equiv t_1 t_2$, where t_1, t_2 are terms of \mathcal{L} , or
2. $\phi \equiv R t_1 t_2 \dots t_n$ where R is an n -ary relation symbol of \mathcal{L} and t_1, t_2, \dots, t_n are all terms of \mathcal{L} , or
3. $\phi \equiv (\neg \alpha)$ where α is a formula of \mathcal{L} , or
4. $\phi \equiv (\alpha \vee \beta)$, where α and β are formulas of \mathcal{L} , or
5. $\phi \equiv (\forall v)(\alpha)$, where v is a variable and α is a formula of \mathcal{L}

Notice that the five clauses of the definition can be separated into two groups. The first two clauses, the atomic formulas, are explicitly defined. The last three clauses are the recursive case, showing how if α and β are formulas, they can be used to build more complex formulas, such as $(\alpha \vee \beta)$ or $(\forall v)(\alpha)$. Now since the collection of formulas is defined recursively, we can use an inductive style proof when we want to prove that something is true about every formula. The inductive proof will consist of two parts, a base case and an inductive case. First we prove the statement for every atomic formula and then using the inductive method we prove it for recursive formulas from the atomic ones. *This method is called induction on the complexity of the formula, or induction on the structure of the formula.*

► **A First-order Language:** Before getting to Sentences one should know a definition for a first-order language. A first-order language \mathcal{L} is defined as an infinite collection of symbols, separated into the following categories:

- *Parentheses:* $(,)$.
- *Connectives:* \wedge, \vee, \neg .
- *Quantifier:* \forall, \exists .
- *Variables:* one for each positive integer n denoted: v_n for n th number.
- *Equality:* $=$.

- *Constant*: We can have a new symbol for each positive number or any other method that we distinguish between two numbers (we can use $|$ for 1, $||$ for 2 etc)
- *Functions*: For each positive integer n , some set of zero or more n -ary function symbols.
- *Relation*: For each positive integer n , some set of zero or more n -ary relation symbols.

Callout — Having n arity means that it is intended to represent a function of n variables.

Notice that by defining such language one can avoid the process of finding an algorithm of grammar (the one that differentiates between nonsense and meaningful words) since we defined all the possible functions and etc. This way we have to only

► **Sentences**: Among the formulas in the language \mathcal{L} , there are some in which we will be especially interested. These are the sentences of \mathcal{L} . The formulas that can be either true or false in a given mathematical model.

Definition 11 — A sentence in a language \mathcal{L} is a formula of \mathcal{L} that contains no free variable.

► **Structures**: Defining any language and with constants, functions and etc. There can exist multiple ways to define structures as opposed to another. The point is that there is no preference. Thus without determining the structure under consideration, without deciding how we wish to interpret the symbols of the language, we have no way of talking about the truth or falsity of a sentence. Thus we have:

Definition 12 — Fix a language \mathcal{L} . An \mathcal{L} -Structure \mathcal{A} is a nonempty set A called the **Universe of \mathcal{A}** , together with:

1. For each constant symbol c of \mathcal{L} , an element $c^{\mathcal{A}}$ of A
2. For each n -ary function symbol f of \mathcal{L} , a function $f^{\mathcal{A}} : A^n \rightarrow A$, and
3. For each n -ary relation symbol R of \mathcal{L} , and n -ary relation $R^{\mathcal{A}}$ on A .

References

- [1] Wikipedia contributors. Mathematical induction — Wikipedia, the free encyclopedia, 2023. [Online; accessed 5-June-2023].