Incompleteness

Çağlar Doğan

CMPE 220 – Fall 2019

Axiomatic Systems

- An axiomatic system is any set of axioms from which some or all axioms can be used in conjunction to logically derive theorems. The oldest examples of axiomatized systems are Aristotle’s syllogistic and Euclid’s geometry.

- By the definition of axiomatic system, one needs to assume certain axioms to make any statement about the truth of other statements.

- Contrary to what most people think, mathematics is not a single axiomatic system. And even further than that, even the fundamental laws of logic are not agreed upon.

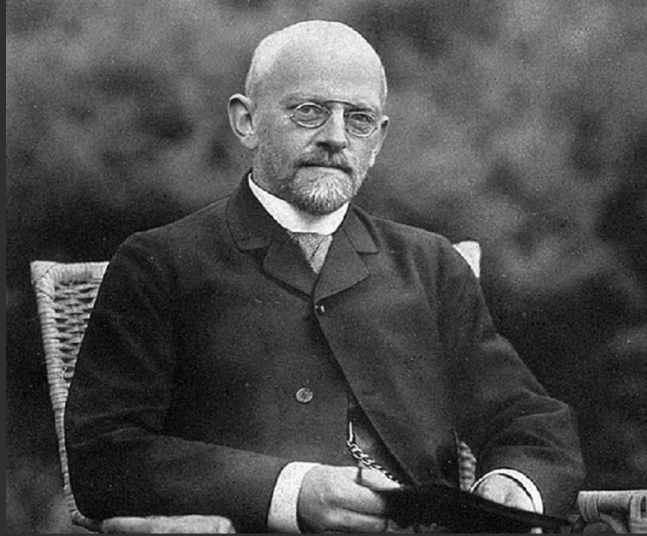
Classical and Intuitionistic Logic

- Classical logic assumes the truth of the three laws of thought:

1. The law of identity (A if and only if A)
2. The law of non-contradiction (A or ¬ A)
3. The law of excluded middle ($(P\to\bot)\to\neg P$)

- Contrary to classical logic, intuitionistic logic does not readily accept the law of excluded middle (although it does not necessarily reject it), rendering some forms of proofs such as proof by contradiction invalid.

Hilbert’s Program



Hilbert’s Program, by German mathematician David Hilbert, was a proposed solution to the **foundational crisis of mathematics** (Grundlagenkrise der Mathematik) which was a search for proper foundations of mathematics.

As a result of the ever-growing paradoxes and inconsistencies linked to the increasing abstraction of mathematics towards the end of 19th century, mathematicians were desperate to find a proper foundations of mathematics to solve these issues in the late 19th and early 20th century.

As a solution for the foundational crisis, Hilbert proposed to construct a complete set of axioms which are proven to be consistent and to ground all existing theories on these axioms. He theorized that once such a set was constructed, the consistency of more complicated systems could be proven by using simpler systems already proven.

Hilbert’s Program

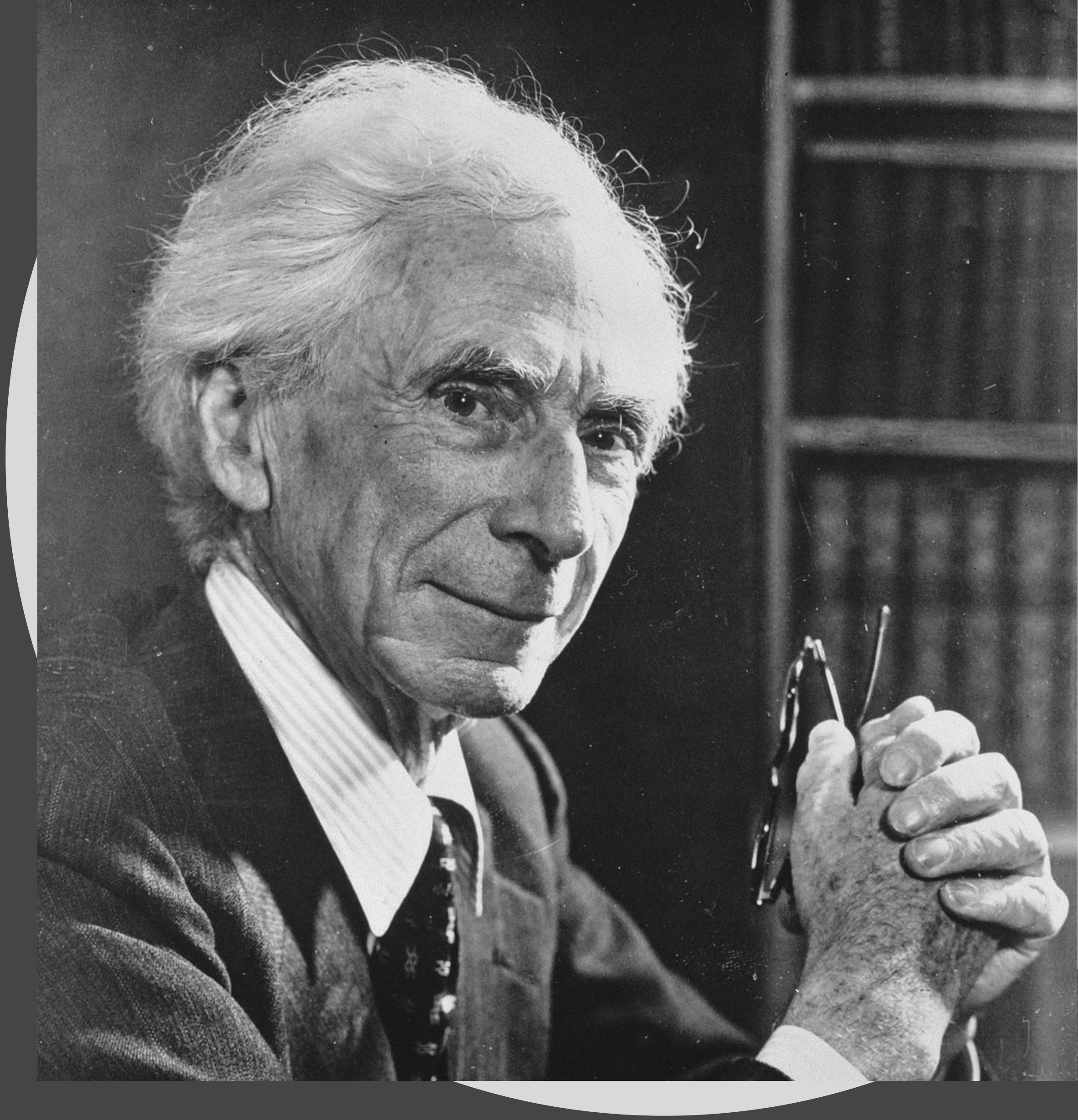
An understanding of 2 particular terms are essential in understanding Hilbert’s Program and the following mathematical concepts:

- Completeness: a proof that all true mathematical statements can be proved in the formalism.
- Consistency: a proof that no contradiction can be obtained in the formalism of mathematics.
- Decidability: existence of an algorithm for deciding the truth or falsity of any mathematical statement. (See: CMPE 350: Formal Languages and Automata Theory)

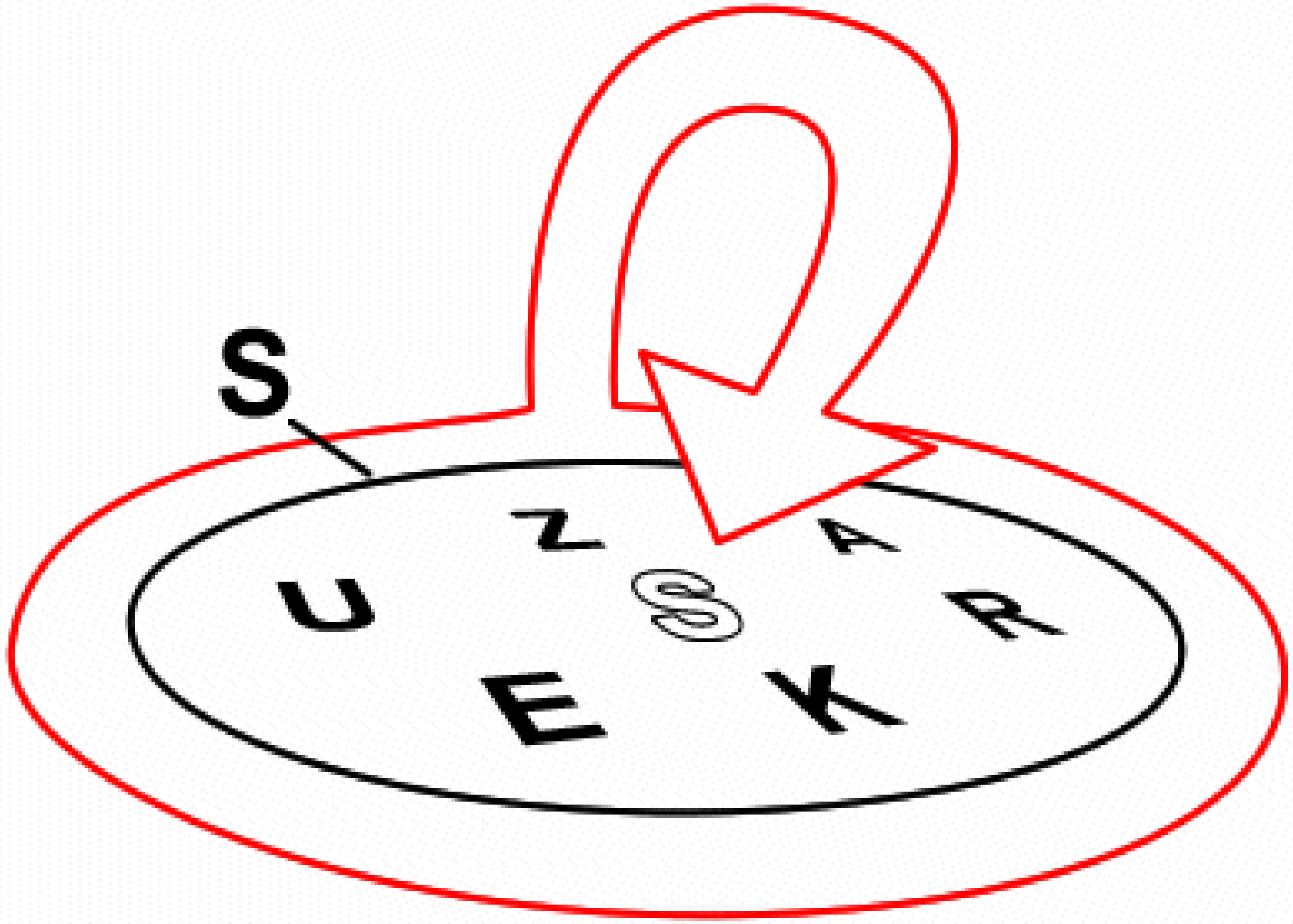
Russell’s Paradox

- “Also known as the Russell-Zermelo paradox, the paradox arises within set theory by considering the set of all sets that are not members of themselves. Such a set appears to be a member of itself if and only if it is not a member of itself. Hence the paradox.”

- https://plato.stanford.edu/entries/russell-paradox/



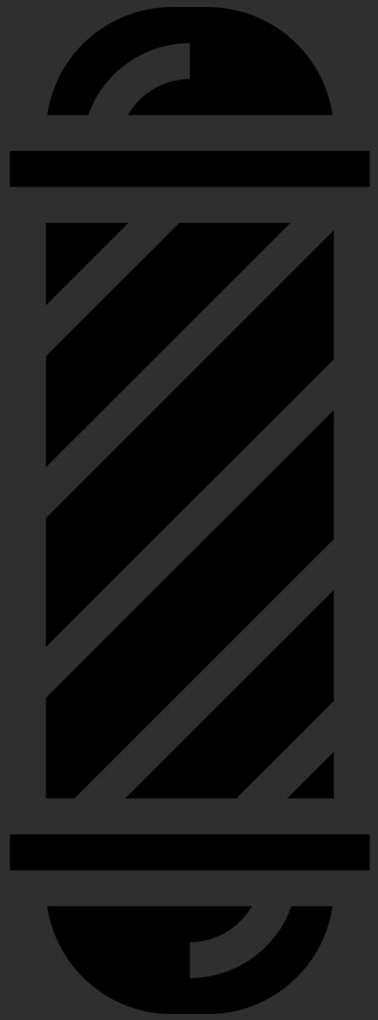
Russell's Paradox asks whether, if S is the set of all sets which do not have themselves as a member, is S a member of itself?



Source: https://tr.pinterest.com/

The Barber Paradox (*Bertrand Russell*)

- One popular representation of Russell’s Paradox is the infamous Barber Paradox by Bertrand Russel.
- The paradox describes a small town where everyone is clean-shaven with only one barber. The barber is tasked with “shaving all those, and those only, who do not shave themselves“. From this seemingly simple definition, an unsolvable question arises: Does the barber shave himself?
- “The barber cannot shave himself as he only shaves those who do not shave themselves. Conversely, if the barber does not shave himself, then he fits into the group of people who would be shaved by the barber, and thus, as the barber, he must shave himself.” - https://en.wikipedia.org/wiki/Barber_paradox



Gödel’s Incompleteness Theorems

1. First Incompleteness Theorem: "Any consistent formal system F within which a certain amount of elementary arithmetic can be carried out is incomplete; i.e., there are statements of the language of F which can neither be proved nor disproved in F." (Raatikainen 2015)

2. Second Incompleteness Theorem: No consistent axiomatic system which includes Peano arithmetic can prove its own consistency.

The proofs are out of the scope of this presentation. A proof sketch can be found here: https://en.wikipedia.org/wiki/Proof_sketch_for_G%C3%B6del%27s_first_incompleteness_theorem

The History of the Incompleteness Theorems

- Gödel actually arrived at the first observations about incompleteness while attempting to *contribute* to Hilbert's program.

- While trying to prove the consistency of analysis, Gödel faced various paradoxes (such as the Liar paradox), and concluded that arithmetical truth cannot be defined in arithmetic.

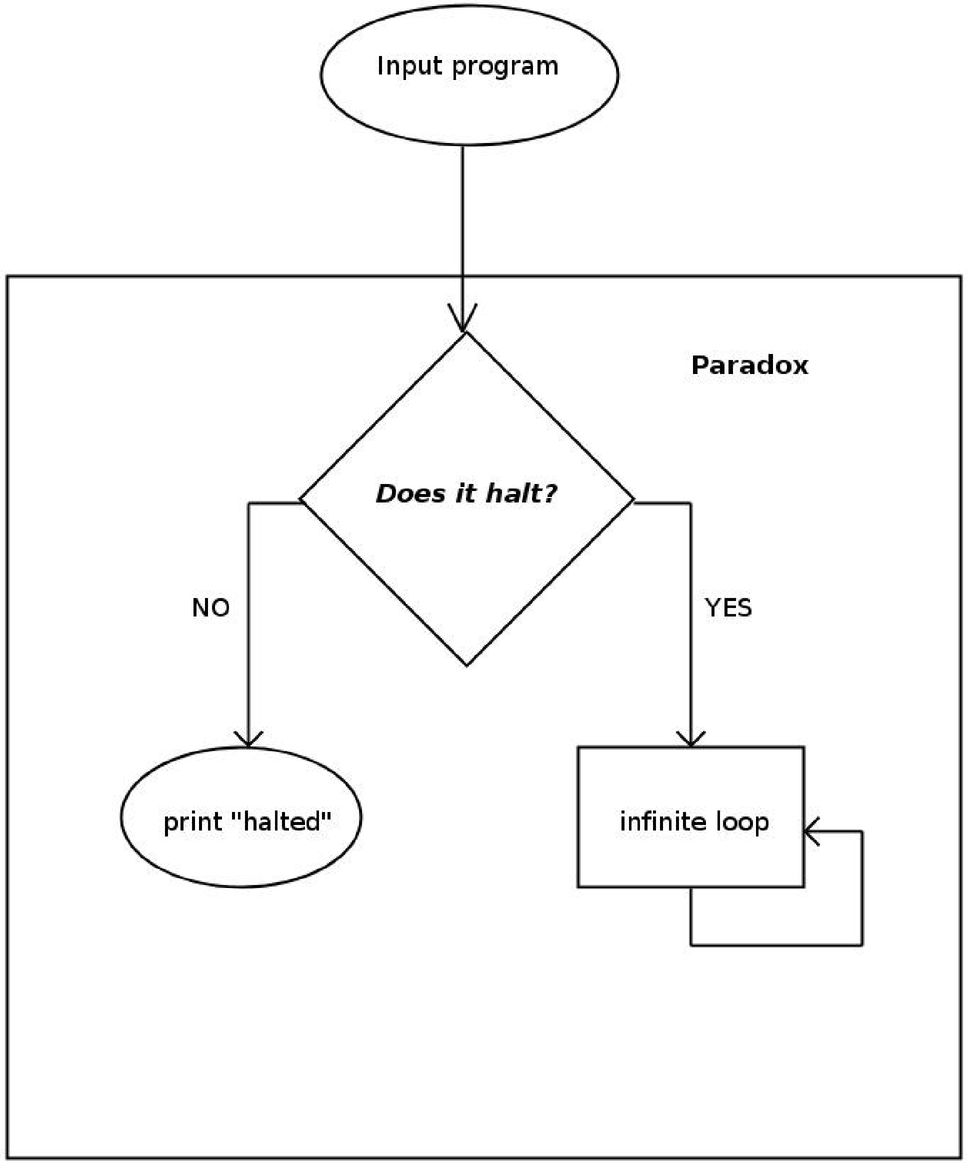
- It should be noted that Gödel was with Vienna Circle with its radically anti-metaphysical attitude. In particular, even the notion of truth was considered as suspicious. Therefore, Gödel eliminated any appeal to the notion of truth and proved incompleteness without truth and consistency. He therefore introduced the notion of ω-consistency. (https://en.wikipedia.org/wiki/%CE%A9-consistent_theory) This led to the incompleteness theorems in the form that they are now known.

- As Gödel's original approach focused on Peano arithmetic and its extensions, some doubted the generality of Gödel's results. In a letter to Gödel July 1932, Alonzo Church suggested that Gödel's results would not apply to his system of λ-conversion. This system was later proved to be inconsistent by Kleene and Rosser.

- “Gödel extended the results to a wider class of systems in papers in 1932 and 1934. He also suggested that his methods would be applicable to standard systems of set theory (however, it was only after the satisfactory characterization of decidability and the Church-Turing thesis a few years later that it was possible to give a fully general formulation of the incompleteness theorem; this was first done in Kleene 1936)”. - https://plato.stanford.edu/entries/goedel-incompleteness/#FirIncThe

Incompleteness and Computer Science

The Halting Problem



- Halting: If a program halts on a certain input, it will never go into an infinite loop.

- The halting problem describes the impossibility of an algorithm which, for every possible program, decides if the program is going to halt on a certain input.
- The impossibility of such a program can be seen by constructing a program P as:

```
If( willHalt(P) ){  //if this program is going to halt
	while(true) //does not halt
		do something;
}
else { //if this program is not going to halt
	Halt; //halts
}
```

Halting Problem and Incompleteness

- The halting problem is closely linked to incompleteness as it presents a solid example of it: The problem essentially demonstrates that there is no computer program that can correctly determine, given any program P as input, whether P will eventually halt.
- In fact, this problem has been used to prove the First Incompleteness Theorem by Shoenfield (1967), Charlesworth (1980), and Hopcroft and Ullman (1979).

- The problem is also very important for computer science as it illustrates that the only sure way of knowing the output for every program is running the program. (And even then, it is impossible to say: “The program will not halt.” at any point. See: Recognizable/Unrecognizable and Decidable/Undecidable Languages in CMPE 350)

Sources

- https://en.wikipedia.org/
- https://plato.stanford.edu/
- https://www.britannica.com/
- http://mathworld.wolfram.com/
- https://math.stackexchange.com/