



La teoría que compone al campo de los Lenguajes formales y los autómatas proviene de un campo que, aparentemente, se encuentra alejado de la Informática: la Lingüística. El primer autor que desarrolló teorías formales sobre gramáticas y lenguajes fue *Noam Chomsky*, considerado una de las figuras más destacadas de la lingüística moderna al abordar temas como el origen y la naturaleza de los lenguajes naturales. A raíz del trabajo de Chomsky, el concepto de *Gramática Formal* adquirió una relevancia importante en la especificación de lenguajes de programación tempranos como **ALGOL 60** (1956) con una gramática libre de contexto.



Figura 1:
Avram Noam
Chomsky
(1928-)

Para el planteamiento formal y abstracto de tales conceptos, hay que remontarse hacia los años 1900, donde el matemático *David Hilbert* se planteó la creación de un sistema matemático formal, completo y consistente en el cual, cada aseveración fuese planteada con precisión. Con esto, su objetivo era hallar un algoritmo que pudiese determinar la verdad o falsedad de cualquier proposición en tal sistema formal.



Figura 2:
David
Hilbert
(1862-1943)

Por medio del planteamiento de este problema, Hilbert definió el *problema de decisión* o *Entscheidungsproblem*. De encontrar Hilbert tal algoritmo, cualquier problema bien definido podría resolverse al ejecutarlo. Sin embargo, tal búsqueda probó ser infructuosa a raíz de los trabajos del matemático Kurt Gödel y su *Teorema de Incompletitud* (1931) el cual formula que *ningún sistema deductivo que contenga a los teoremas de la aritmética, y cuyo conjunto de axiomas sea recursivo, no es consistente y completo a la vez*.

La computación, al tener sus bases fundamentadas en sistemas matemáticos como la aritmética, tiene sus limitaciones. A raíz de lo anterior han surgido problema conocidos como *indecidibles*, entre los cuales se encuentran:

- **Entscheidungsproblem:** El problema de decisión planteado originalmente por Hilbert fue probado como indecidible por el matemático británico *Alan Turing* y el matemático estadounidense *Alonzo Church* de forma independiente.
- **El problema de la parada:** Este problema plantea que, dado un programa y su entrada, decidir si tal programa terminará para tal entrada o se ejecutará indefinidamente. Alan Turing demostró que tal problema recae en lo indecidible.
- **La computabilidad de los números:** Un número computable se trata de un número real que puede ser aproximado por un algoritmo con un nivel arbitrario de exactitud. Turing demostró que la gran mayoría de números reales no son computables. Un ejemplo notable se trata de la *Constante de Chaitin* $[\Omega]$.



Figura 3:
Kurt
Gödel(1906-
1978)

Los trabajos de Kurt Gödel tuvieron una influencia notable en el concepto de la *computabilidad* y el planteamiento de los problemas que involucran el uso de *Máquinas de Turing*, un dispositivo abstracto que manipula símbolos sobre una tira de cinta de acuerdo con una tabla de reglas propuesto por Alan Turing en 1937 en su artículo *Sobre los números computables*. Este ente abstracto permitió validar las ideas de Gödel respecto a la computabilidad matemática, y además, funcionó como el ente precursor de las primeras computadoras contruidas por la humanidad.

Los lenguajes formales que son aceptados por una máquina de Turing son aquellos que pueden generarse por una gramática formal. Por ejemplo, el cálculo Lambda permite definir por medio de expresiones funciones computables de igual forma por una máquina de Turing de tal forma que resultan equivalentes. Tal conexión permite enlazar el campo de los lenguajes con la computabilidad; una importante conexión que dio lugar al desarrollo de los lenguajes de programación modernos y permitió comprender de mejor forma los alcances y las limitaciones que la computación posee.

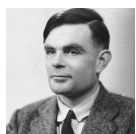


Figura 4:
Alan
Turing(1912-
1954)