Lógica Superior:

# Computabilidad e Incompletitud

Profesor: Eduardo Barrio
UBA - Filosofía
2do cuatrimestre de 2015

# Bibliografía:

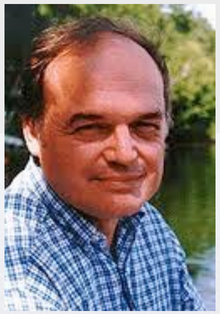
**Manual:**

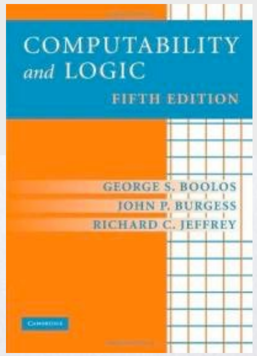
Boolos, G., Burgess, J. & Jeffrey, R. *Computability and Logic.* Cambridge, New York: Cambridge Caps. 1, 2, 3, 4, 6 , 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 y 18.

**Consulta:**

Smith, P. *An Introduction to Gödels Theorems.* (Cambridge, Cambridge UP, 2009).

Hunter, G. *Metalógica* Madrid, Paraninfo. 1981. Segunda Parte.





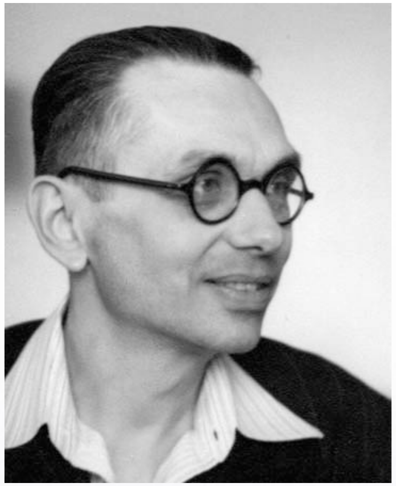
# Estructura del Curso

- **Clases Teóricas**: se destinarán a la exposición de los principales temas del presente programa. Cuatro horas semanales: lunes y miércoles de 17 a 19 hs.
- **Clases Prácticas:** lunes y miércoles de 15 a 17 hs.
- Evaluación: dos parciales presenciales a libro abierto.
- **Sitio: BA-Logic: http://www.ba-logic.com/logica-superior**

# Objetivos

1. - Analizar los vínculos entre lógica, aritmética y computabilidad.
2. - Comprender los principales problemas y motivaciones para construir los sistemas de lógica de primer orden.
3. - Demostrar los metateoremas de Corrección y de Completitud de la lógica clásica de primer orden.
4. - Demostrar, utilizando diversas técnicas, los metateoremas de Compacidad y Löwenheim-Skolem.
5. - Demostrar los Teoremas de Gödel (Primer y Segundo Teorema de Gödel) y analizar sus consecuencias filosóficas.
6. - Demostrar los teormemas de Tarski y de Löb y analizar sus consecuencias filosóficas.

# Protagonistas:



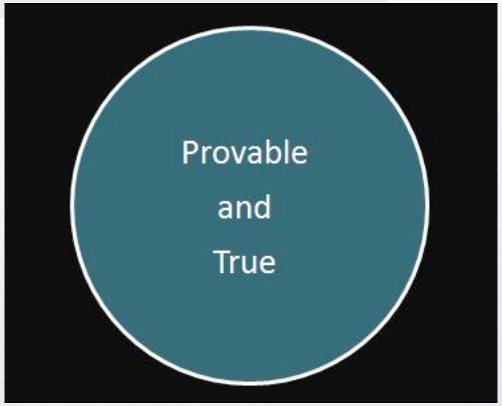

# *Interés por los fundamentos de la matemática:*

Dos grandes objetivos (Hilbert)

1. **Completitud** de la aritmética

se buscaba un sistema axiomático que capturara **todas** las verdades de la aritmética

2. **Decisión** de las teorías (Entscheidungsproblem)
se buscaba un **procedimiento efectivo** (algoritmo) para decidir si cualquier fórmula de primer orden era válida o no



# **Teoría Aritmética:** presentación axiomática informal

**Axiomas en el lenguaje natural:**

1) Cero es un número natural.

2) El sucesor inmediato de un número natural es un número natural

3) Cero no es el sucesor inmediato de un número natural

4) No hay dos números naturales que tengan el mismo sucesor inmediato.

5) Si una propiedad se aplica a cero y dado cualquier número, si se aplica a él, también se aplica a su sucesor inmediato, entonces esa propiedad se aplica a todos los números naturales.

# **Teoría Aritmética:** presentaciones axiomáticas formales

***Q*** (Robinson Arithmetic)

Axioma 1: $\forall x\ (0 \neq Sx)$

Axioma 2: $\forall x \forall y\ (Sx = Sy \rightarrow x=y\ )$

Los dos primeros axiomas definen las características mínimas acerca de los números naturales (que es un conjunto infinito con un primer elemento).

Axioma 3: $\forall x\ (x \neq 0 \rightarrow \exists y\ (x = Sy)\ )$

Asegura que haya un solo cero: evita dominios con seudo-ceros. Cuando se tiene el axioma de inducción, Ax 3 es un teorema de PA.

Definición recursiva de adición

Axioma 4: $\forall x\ (x+0 = x)$

Axioma 5: $\forall x \forall y\ (x+Sy = S\ (x+y))$ Definición recursiva de multiplicación

Axioma 6: $\forall x\ (x \times 0 = 0)$

Axioma 7: $\forall x \forall y\ (x \times Sy = (x \times y)+x)$

# **Teoría Aritmética:** presentaciones axiomáticas formales

**PA** (Peano Arithmetic)

Axioma 1: $\forall x\,(0 \neq Sx)$

Axioma 2: $\forall x \forall y\,(Sx = Sy \rightarrow x=y\,)$

Axioma 3: $\forall x\,(x+0 = x)$

Axioma 4: $\forall x \forall y\,(x+Sy = S\,(x+y))$

Axioma 5: $\forall x\,(x \times 0 = 0)$

Axioma 6: $\forall x \forall y\,(x \times Sy = (x \times y)+x)$

Axioma de Inducción:
$(\varphi\,(0) \wedge \forall x(\,\varphi(x) \rightarrow \varphi(Sx))) \rightarrow \forall x\,(\varphi(x))$

donde “$\varphi(x)$” se puede reemplazar por alguna fórmula abierta de LA con “x” libre. Deseamos que “$\varphi(x)$” exprese genuinas propiedades (propiedades que caen realmente bajo el alcance del principio de inducción).

# **Teoría Aritmética:** presentaciones axiomáticas formales

***$PA^2$*** (Second Order Peano Arithmetic)

Axioma 1: $\forall x\ (0 \neq Sx)$

Axioma 2: $\forall x \forall y\ (Sx = Sy \rightarrow x=y\ )$

Axioma 3: $\forall x\ (x+0 = x)$

Axioma 4: $\forall x \forall y\ (x+Sy = S\ (x+y))$

Axioma 5: $\forall x\ (x \times 0 = 0)$

Axioma 6: $\forall x \forall y\ (x \times Sy = (x \times y)+x)$

Axioma de Inducción con cuantificación de segundo orden:

$\forall X\ ((X0 \wedge \forall x(Xx \rightarrow XSx)) \rightarrow \forall x\ Xx)$

El esquema de inducción usado en ***$PA^1$*** se deriva de axioma de inducción de ***$PA^2$.***

# Resultados metalógicos:

**1.-** Una teoría T es correcta ssi todo teorema de T es verdadero.

**2.-** Una teoría T es decidible ssi la propiedad de ser un teorema en T es una propiedad efectivamente decidible. Hay un procedimiento mecánico para determinar, para cualquier oración de T, si esa oración es un teorema en T.

**3.** - Una teoría T decide la oración $\alpha$ ssi $\alpha$ es un teorema en T o $\neg\alpha$ es un teorema en T.

**4.**- Una teoría T decide correctamente la oración $\alpha$ ssi si $\alpha$ es verdadera, $\alpha$ es un teorema en T y si $\alpha$ es falsa $\neg\alpha$ es un teorema en T.

**5.**- Una teoría T es completa respecto de la negación ssi T decide toda oración $\alpha$ de su L. Es decir, para cualquier $\alpha$, o es un teorema en T $\alpha$ o su negación.

$$\vdash \alpha \quad \text{o} \quad \vdash \neg\alpha$$

**6.**- Una teoría T es inconsistente ssi para alguna oración $\alpha$ , se puede probar en T $\alpha$ y $\neg\alpha$.

$$\vdash \alpha \quad \& \quad \vdash \neg\alpha$$

# Resultados metalógicos:

Idealmente, quisiéramos tener una teoría (como Q o PA) que sea capaz de probar todas las verdades expresables en el lenguaje de la aritmética. Deseamos una T completa respecto de la negación.

Que una T sea (semánticamente) completa es distinto a decir que sea completa respecto de la negación.

# Relación entre completitud y decibilidad:

El conjunto de los teoremas de T puede ser efectivamente enumerado. Es decir, podemos dar un algoritmo para enumerar mecánicamente las pruebas de sus teoremas.

- El que T sea efectivamente enumarable no quiere decir que T sea decidible: Una cosa es tener un método mecánico para generar eventualmente cualquier teorema y otra es tener un método mecánico para, dada cualquier oración arbitraria $\alpha$ poder determinar sin continuar para siempre si esa oración está en la lista de teoremas.

- Que una T sea decidible no quiere decir que sea completa respecto de la negación.

- Una cosa es tener un modo mecánico de decidir lo que es un teorema (decidibilidad) y otra es tener recursos suficientes como para probar $\vdash$ o disprove $\nvdash$ toda fórmula del lenguaje

# Autorreferencia en la aritmética

Los lenguajes de primer orden son capaces de formalizar predicados como:

1. Nombrar las fórmulas por medio de códigos (Números de Gödel)
2. For: (fórmula):
3. Bew (predicado de prueba): x es una demostración de y
4. Con (predicado de consistencia): x es consistente
5. Tr (predicado veritativo): x es verdadera

# Primer Teorema de Gödel

Gödel (1931): cualquier sistema axiomático suficientemente poderoso es incompleto o inconsistente

Estrategia de la Prueba:

- Tesis: En *PA* hay fórmulas verdaderas que no pueden probarse.

Encontrar un caso *G:* “Yo no soy demostrable”. $G \leftrightarrow \neg \mathrm{Bew}(G)$

- La fórmula *G* de Gödel : Esta fórmula de la teoría de números no tiene ninguna prueba en *PA*

- *G* no tiene una prueba, pero es verdadera!

1. - Si *G* fuera probable, entonces *PA* sería inconsistente.
2. - Si *G* fuera no probable, entonces *PA* sería incompleto.
3. - Por eso, *PA* no puede ser completo y consistente!

# Segundo Teorema de Gödel

Si *PA* es consistente, entonces *PA* no puede probar como teorema una fórmula que afirme su propia consistencia.

PA $\nvdash$ Con(PA),

No hay ninguna teoría más débil que *PA* que pueda demostrar la consistencia de PA. ¿Qué pasa con el programa de Hilbert?

- Gödel demostró que es imposible establecer la consistencia lógica interna de una amplia clase de sistemas deductivos (dentro de los que se incluye la aritmética), a menos que se adopten principios tan complejos de razonamiento que su consistencia interna quede tan sujeta a la duda como la de los propios sistemas.

# El problema de la decisión (Entscheidungsproblem)

Se buscaba un procedimiento efectivo para decidir si cualquier fórmula de primer orden era válida o no

Un procedimiento es computable si se puede especificar una secuencia de instrucciones (procedimiento **efectivo / algoritmo)** las cuales cuando son seguidas (mecánicamente) permiten completar el procedimiento en un tiempo finito.

*Noción formal de computación: Turing-computability*

***Tesis Turing-Church:*** la noción de *Turing-computability* captura exactamente la idea intuitiva *de procedimiento efectivo.*

# Formulaciones alternativas de Computabilidad

Funciones Recursivas

Las funciones Turing computables son exactamente las funciones recursivas.

Turing (1936): no existe tal procedimiento efectivo

- Toda formulación axiomática consistente de la teoría de números incluye fórmulas indecidibles.
- Fórmula indecidible: no puede probarse o su verdad o su falsedad dentro de la teoría de números.