Introducción

Clase 00 - Lógica para Ciencia de la Computación/Teoría de la Computación

Prof. Miguel Romero

Objetivo del curso:

Introducir los elementos básicos de la teoría de la computación.

Programa

- Unidad I: Computabilidad
- Unidad II: Lógica de primer orden
- Unidad III: Complejidad Computacional

- ¿Qué problemas pueden ser resueltos por un algoritmo?
- ¿Cómo definimos la noción de algoritmo?

- Ecuaciones de polinomios (en varias variables) con coeficientes enteros.
- Buscamos soluciones enteras.

Ejemplos

$$3x^2 - 2xy - y^2z = 7 \qquad \text{(tiene solución } x = 1, y = 2, z = -2\text{)}.$$
$$x^2 + y^2 = -1 \qquad \text{(no tiene solución)}.$$
$$6x + 9y = 11 \qquad \text{(no tiene solución)}.$$

Problema:

Dada una ecuación diofántica, decidir si tiene solución o no.

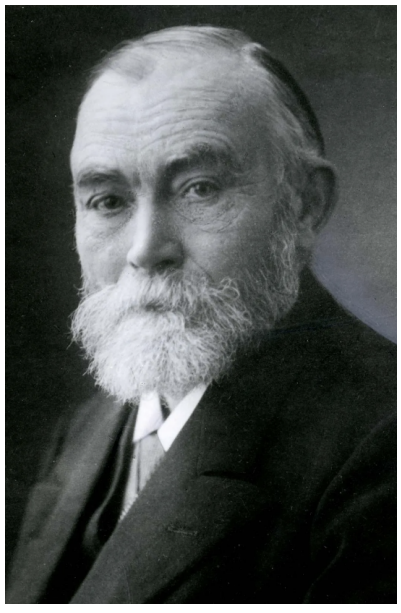
Los 23 problemas de Hilbert (1900)



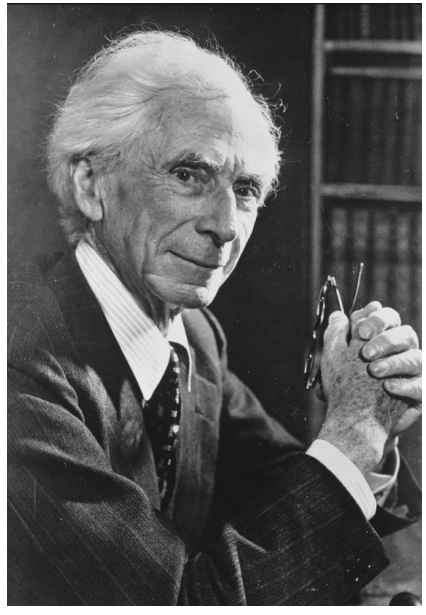
David Hilbert (1862 - 1943)

Décimo problema: Ecuaciones diofánticas.

Crisis fundacional de las matemáticas



Gottlob Frege (1848 - 1925)



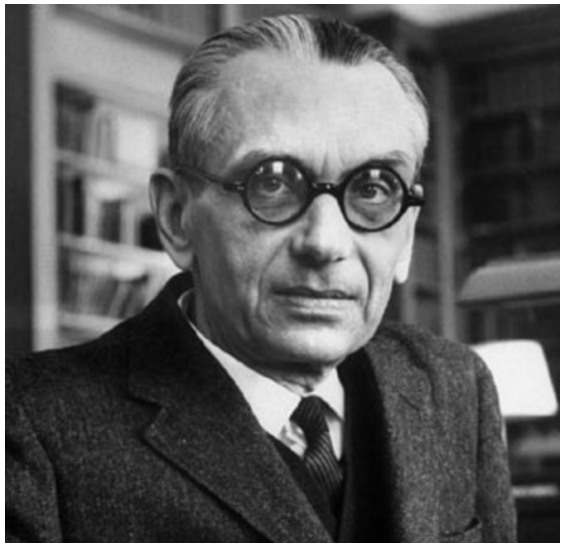
Bertrand Russell (1872 - 1970)

Frege (1879): Sistema formal, basado en lógica y teoría de conjuntos.
Russell (1902): El sistema de Frege es inconsistente (paradoja de Russell).

Programa de Hilbert (1922-1930):

Encontrar un sistema formal para las matemáticas y demostrar que:

- El sistema es **consistente**: no se puede demostrar una proposición matemática y su negación, al mismo tiempo.
- El sistema es **completo**: toda proposición matemática puede ser demostrada o refutada.
- El sistema es **decidible**: Existe una forma mecánica de determinar si una proposición matemática en el sistema es verdadera o falsa.



Kurt Gödel (1906 - 1978)

Teorema de incompletitud (1931):

No existe sistema formal consistente que sea completo.

Entscheidungsproblem (El problema de decisión) [Hilbert-Ackermann 1928]:
Dada una fórmula en la lógica de primer orden, decidir si es válida, es decir, si es siempre verdadera.



Alonzo Church (1903 - 1995)



Alan Turing (1912 - 1954)

Church y Turing (1936):
El problema de decisión es **indecidible**: no tiene solución algorítmica.

Fuerte conexión entre lógica y computación:

- La Ciencia de la Computación nace a partir del estudio de la lógica.

La lógica ha encontrado diversas aplicaciones en Ciencia de la Computación:

- Inteligencia Artificial
- Bases de Datos
- Verificación formal
- Teoría de la Computación
- Lenguajes de Programación
- ...

"El cálculo de la Ciencia de la Computación"

Programa

- Unidad I: Computabilidad
- Unidad II: Lógica de primer orden
- Unidad III: Complejidad Computacional

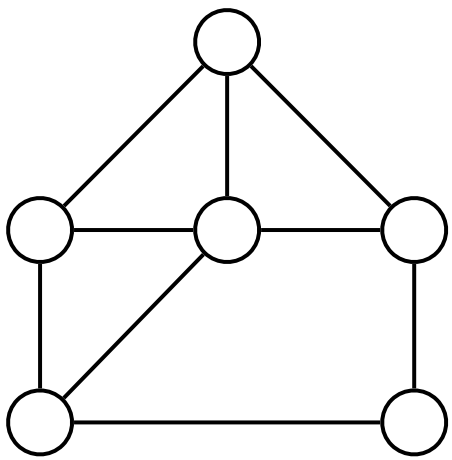
- ¿Qué problemas pueden ser resueltos eficientemente?

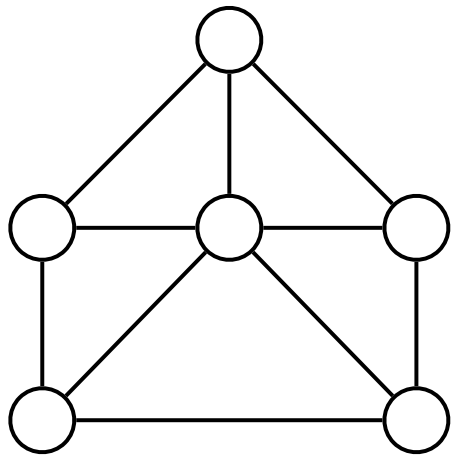
Un grafo no dirigido $G = (V, E)$ es k-**coloreable** si **existe** una función $c : V \to \{1, \ldots, k\}$ tal que:

$$\text{para cada } (u, v) \in E, \text{ se tiene que } c(u) \neq c(v).$$

Es decir: nodos adyacentes recibe colores distintos.

Ejemplo: ¿Cuáles de estos grafos tienen un 3-coloreo?



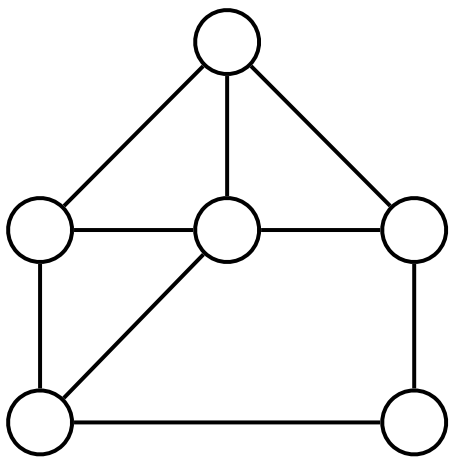


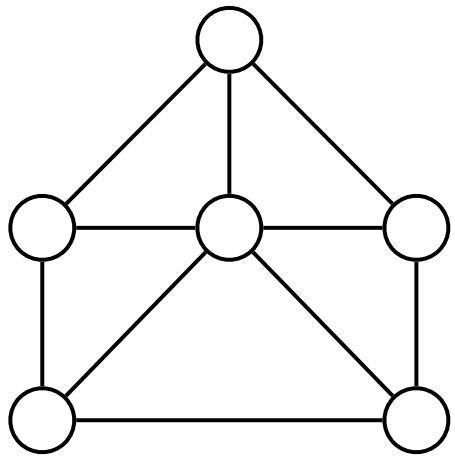
Un grafo no dirigido $G = (V, E)$ es k-**coloreable** si **existe** una función $c : V \to \{1, \ldots, k\}$ tal que:

$$\text{para cada } (u, v) \in E, \text{ se tiene que } c(u) \neq c(v).$$

Es decir: nodos adyacentes recibe colores distintos.

Ejemplo: ¿Cuáles de estos grafos tienen un 2-coloreo?





Problema de 3-coloreo:

Dado un grafo G, decidir si es 3-coloreable.

¿Puede dar un algoritmo para resolver este problema?

¿Cuántas operaciones hace su algoritmo con respecto a la cantidad de nodos n?

Problema de 2-coloreo:

Dado un grafo G, decidir si es 2-coloreable.

¿Puede dar un algoritmo más eficiente para resolver este problema?

P vs NP

(https://www.claymath.org/millennium/p-vs-np/)

Temas administrativos

(ver programa en Canvas)

Cuerpo docente

- Profesor: Miguel Romero (mgromero@uc.cl)
- Ayudante coordinador: Benjamín Álvarez (balvarezs@uc.cl)
- Ayudantes cátedra:
  Caetano Borges (caetano.borges@uc.cl) y Valentina Juri (vjuri@uc.cl)
- Correctores:
  - Mateo Bozzi (mbozzi@uc.cl)
  - Vicente Cabra (vicente.cabra@uc.cl)
  - Gonzalo Matus (gonzalo.matus@uc.cl)
  - Jeremy Peñaloza (jeremy.pealoza@uc.cl)
  - Matias Poblete (mapobletef@uc.cl)
  - Juan Domingo Veloso (jdomingo.veloso@uc.cl)

Se asumen nociones básicas de matemáticas discretas y algoritmos:

- Métodos de demostración: demostraciones directas, demostraciones por contradicción, demostraciones por casos, demostraciones por inducción, ...
- Conceptos básicos: Conjuntos, relaciones, funciones, lógica proposicional, grafos, notación asintótica, ...

Metodología

- 2 clases a la semana (Martes y Jueves módulo 5, sala A5)
- 1 ayudantía a la semana (Viernes módulo 4, sala B13)
- Trabajo personal

Evaluaciones

- 6 tareas
- 2 interrogaciones escritas
- Examen final escrito

Tareas

	Publicación enunciado	Entrega (23:59 hrs)
Tarea 1	Jueves 12 de marzo	Lunes 23 de marzo
Tarea 2	Jueves 26 de marzo	Miércoles 08 de abril
Tarea 3	Martes 21 de abril	Lunes 04 de mayo
Tarea 4	Martes 05 de mayo	Viernes 15 de mayo
Tarea 5	Lunes 01 de junio	Viernes 12 de junio
Tarea 6	Lunes 15 de junio	Jueves 25 de junio

La entrega es via Canvas y **debe** ser escrita en LaTeX.

- Las tareas son **individuales**.
- Puede discutir con sus compañeros, con el equipo docente o utilizar recursos externos (libros, foros, páginas web, IAs).
- No se acepta *copy-paste* de recursos externos ni IAs.

Nota de Tareas (**NT**):

Promedio de las 5 mejores tareas.

5 Cupones `#problemaexcepcional`:

- Cada cupón permite extender el plazo de entrega de una tarea en **1 día**, debido a motivos excepcionales, sin necesidad de una justificación.
- Para cada tarea puede usar **a lo más 3** cupones.

Interrogaciones y examen

I_1	:	Miércoles 15 de abril,	17:30 hrs.
I_2	:	Sábado 30 de mayo,	08:20 hrs.
Examen	:	Martes 30 de junio,	13:30 hrs.

La inasistencia a alguna de las dos interrogaciones no requiere justificación.
No se puede faltar a ambas interrogaciones.
La inasistencia al examen se debe justificar ante la Dipre (Nota P).

Nota de pruebas (**NP**):

$$\mathbf{NP} = \frac{I1 + I2 + 2 \cdot Examen - \min\{I1, I2, Examen\}}{3}$$

Nota final (**NF**):

$$\mathbf{NF} = 0.2 \cdot \mathbf{NT} + 0.8 \cdot \mathbf{NP}$$

El curso se aprueba $\iff$ $\mathbf{NT} \geq 3.95$ y $\mathbf{NP} \geq 3.95$.

En caso de no aprobar, la nota final es $\min\{\, \mathbf{NF}, 3.9 \,\}$.

Canvas

Bibliografía

Recomendados:

- Michael Sipser. *Introduction to the Theory of Computation*. 3ra Edición, 2012.
- L. Bertossi. *Lógica para Ciencia de la Computación*. Ediciones UC, 1996.

Otra bibliografía interesante:

- Harry Lewis, Christos Papadimitriou. *Elements of the Theory of Computation*. 2da Edición, 1997.
- H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 2da edición, 2000.
- Sanjeev Arora, Boaz Barak. *Computational Complexity: A Modern Approach*. 1ra Edición, 2009.