Algebra 1 – Programa

1: Conjuntos, relaciones y funciones:
Conjuntos: definiciones, pertenencia, contenciones, operaciones (unión, intersección, diferencia). Leyes de De Morgan. Cardinal de conjuntos finitos. Tablas de verdad y relación con lógica proposicional. Igualdad de conjuntos (diagramas de Venn, tablas). Producto cartesiano. Conjunto de Partes (y su cardinal para cjtos finitos).
Relaciones: definición, su representación como grafos. Relaciones de orden y equivalencia. Clases de equivalencia. Clausura transitiva.
Funciones: Definición. Composición. Funciones inyectivas, sobreyectivas, biyectiva, inversa.
Cuantificadores: noción intuitiva.

Taller: Introducción al manejo de las computadoras del laboratorio. Introducción a lenguaje funcional: idea de que todo es una función. Haskell: Instalación y entorno de programación (grabar, cargar, etc.). Definición de funciones simples sin recursión. Tipos básicos: Integer, Bool, Char. Definición por casos. If then else. Operaciones aritméticas, evaluación de fórmulas de la aritmética. Operaciones booleanas. Evaluación de fórmulas de la lógica proposicional solo con constantes True y False. Currificación. Producto cartesiano vs. currificación. Composición de funciones. Evaluación parcial. Evaluación eager vs. Lazy.

2: Números naturales e Inducción:
Definición "intuitiva" de los números naturales, primeras demostraciones por inducción (simple). Sumatoria, productoria y su escritura como ciclos en un programa. Factorial y su interpretación combinatoria (biyecciones en conjuntos finitos). Número combinatorio y su interpretación combinatoria (subconjuntos en un conjunto finito), escritura como suma de dos combinatorios, definición recursiva del combinatorio. Definición de funciones recursivas en pseudocódigo (o código en algún lenguaje concreto). Definición por los axiomas de Peano de los números naturales. Ejemplos de demostración por inducción global. Ejemplos de algoritmos recursivos (sort, Hanoi, Fibonacci) y análisis de complejidad. Cálculo de a^n por distintos algoritmos (introducción intuitiva de noción de complejidad). Inducción global y principio de buena ordenación.

Taller: Recursión. Sumatoria y productoria con y sin filtros (por ejemplo, la suma de todos los naturales <= n que cumplan con el predicado P). Definición recursiva de factorial, números combinatorios, etc**.** Tipos algebraicos en Haskell. Pattern matching. Tipo algebraico 1: Naturales como cero y sucesor. Definición de la suma, producto, etc. Tipo algebraico 2: Listas. Recorrido de listas. Búsqueda de un elemento, palíndromo, reverso, etc. Búsqueda de patrones en listas. Lista de listas. Tipo algebraicos 3: fórmula de la lógica proposicional. Programación de satisfacibilidad y validez. Introducción temprana de P y NP. Sorting de listas. Fibonacci: algoritmo directo (ineficiente) y con programación dinámica (eficiente). Tipo algebraico 4: Árboles binarios. Recorrido sobre árboles binarios: BFS, DFS.

3: Números enteros:
Enteros. Divisibilidad y primeras propiedades. Primos y Compuestos. Algoritmo de división. Aplicaciones del algoritmo de división. Escrituras en distintas bases, sistemas de numeración. Máximo común divisor. Algoritmo de Euclides (y su complejidad), escritura del máximo común divisor como combinación lineal. Numeros coprimos. Propiedades. Teorema Fundamental de la aritmética. Cantidad de primos. Criba. Aplicaciones del TFA (cantidad de divisores, cálculo de gcd y del mcm). Curiosidades de los primos. Congruencias, propiedades y aplicaciones (criterios de

divisibilidad). Restos modulo m. Grupos y Anillos (comparación de Z/mZ con Z). Ecuaciones lineales diofánticas y ecuaciones de congruencia. Algoritmos. Sistemas de ecuaciones de congruencia. Teorema Chino del Resto. Pequeño Teorema de Fermat. Algoritmos probabilisticos de primalidad. de Euler-Fermat. Aplicación: Algoritmo criptográfico RSA.

Taller: Algoritmo de División. Programación (funcional) del algoritmo de Euclides y cálculo de su complejidad. Descomposición en factores primos (calcular lista de divisores). Criba de Eratóstenes. Programación de criterios de divisibilidad con enteros representados como listas de dígitos decimales. Programación de resolución de ecuaciones diofánticas lineales. Programación del teorema Chino del Resto: resolución de sistemas de congruencias. Números pseudo aleatorios. Programación de algoritmos probabilísticos de primalidad u otros algoritmos probabilísticos. Logaritmo discreto.

4: Polinomios con coeficientes en un cuerpo:

Cuerpos. Definición y ejemplos, Q, R, C, Z/pZ. Anillo de polinomios K[x]: generalidades (suma, producto, unidades), grado, divisibilidad, irreducibles y compuestos, algoritmo de división. Paralelismo con Z: Máximo común divisor, algoritmo de Euclides, coprimos. Factorización única.
Aspecto funcional: Evaluacion de polinomios (def y algoritmos). Raíces. Teorema del resto. Resolución de cuadraticas en K[X]. Multiplicidad. Equivalencias. Cota para el número de raíces con multiplicidad sobre un cuerpo.
C[X]: Repaso del cuerpo C, coordenadas polares, fórmulas de Moivre. Raíces/factorizacion de X^n – z en C[x]. Grupo de raíces de la unidad. Teorema Fundamental del Algebra, irreducibles de C[X].
R[X]: Raíces complejas no reales de polinomios reales. Factorización en R[X].
Q[X]: Teorema de Gauss para calcular raíces racionales.
Ejemplos de factorización en K[X] para distintos K. Criterios de irreducibilidad sobre Q y algoritmos de factorización sobre los distintos cuerpos

Taller: Tipo algebraico 5: complejo. Programación de operaciones básicas. Representación de polinomios. Operaciones básicas con las diferentes representaciones. Complejidad de tiempo y espacio de cada una. Ejemplos de programación imperativa. Algoritmos para encontrar raíces en polinomios. Algoritmo ingenuo de factorización sobre Q. Más ejemplos de programación imperativa.

Bibliografía

- E. Gentile. Notas de Algebra. EUDEBA,
- E. Gentile. Estructuras algebraicas I. Monografía científica de la OEA, 1977.
- E. Gentile. Aritmética Elemental. Monografía científica de la OEA, 1985.
- G. Birkhoff, S. Mc Lane. Algebra moderna. Vicens-Vives (4ta ed.), 1970.
- Richard S. Bird, Philip L. Wadler . An Introduction to Functional Programming. Prentice-Hall, 1988.
- Richard Bird. Introduction to Functional Programming Using Haskell. Prentice-Hall, 1998.
- Edsger Dijkstra. A Discipline of Programming. Prentice Hall, 1997.
- Conjuntos, relaciones y funciones, por Susana Puddu
  http://cms.dm.uba.ar/academico/materias/1ercuat2012/algebra_I/Conjuntos-Puddu.pdf

- Números naturales, principio de inducción, por Susana Puddu
  http://cms.dm.uba.ar/academico/materias/1ercuat2012/algebra_I/Naturales-Puddu.pdf
- Combinatoria, por Susana Puddu
  http://cms.dm.uba.ar/academico/materias/1ercuat2012/algebra_I/Combinatoria-Puddu.pdf
- Números enteros, por Susana Puddu
  http://cms.dm.uba.ar/academico/materias/1ercuat2012/algebra_I/Enteros-Puddu.pdf
- Números enteros, por Teresa Krick.
  http://cms.dm.uba.ar/academico/materias/1ercuat2012/algebra_I/enteros2011-4.pdf
- Números complejos, por Susana Puddu
  http://cms.dm.uba.ar/academico/materias/1ercuat2012/algebra_I/Complejos-Puddu.pdf
- Polinomios, por Susana Puddu
  http://cms.dm.uba.ar/academico/materias/1ercuat2012/algebra_I/Polinomios-Puddu.pdf
- Notas de Ariel Pacetti y Matías Graña
  http://cms.dm.uba.ar/academico/materias/2docuat2012/algebra_I/main.pdf