C 2006
(39)

ESPECIFICACIÓN Y COMPLEJIDAD EN EL CÁLCULO CIENTÍFICO. EL ARTE

11.- PROGRAMA ANALÍTICO (adjuntarlo).

Objetivos

El curso pretende introducir a los alumnos al análisis y construcción de programas complejos mediante técnicas de verificación y derivación de programas ejecutables para el cálculo científico a partir de especificaciones formales. Se ilustrará con numerosos ejemplos como especificaciones descriptivas, formuladas en términos provenientes de la teoría de bases de datos, se convierten en especificaciones operacionales y finalmente en algoritmos sujetos a diferentes técnicas de optimización. En su aspecto práctico, el curso enseñará, de forma independiente de un lenguaje determinado, algunas técnicas de especificación y derivación de programas y algunos tipos y estructuras de datos y algoritmos básicos y eficientes para el cálculo simbólico y numérico. En su parte teórica, el curso tratará de discutir la relación entre programa (con verificación) y algoritmo, haciendo hincapié en los aspectos de complejidad. El curso desarrollará una visión unificada de la teoría de la complejidad de algoritmos, de la especificación y transformación de programas y de la teoría de bases de datos (continuas y relacionales). Todos los conceptos serán introducidos mediante de la discusión de problemas concretos provenientes principalmente de la geometría algorítmica y desarrollados hasta su formulación matemática rigurosa.

Temática

La computación científica exige con frecuencia del desarrollo de programas complejos cuyo funcionamiento práctico requiere técnicas razonadas de derivación y verificación y un estrecho control de la complejidad de los algoritmos subyacentes. A su vez, la eficiencia de los algoritmos implementados depende de la elección de los tipos y estructuras de datos que se utilizan. Esto conduce al concepto de especificación: en la práctica es imposible redactar un programa complejo en un lenguaje de bajo nivel. Tal programa sería fácilmente incorrecto, ineficiente, poco flexible y la tarea de programación no sería distribuible entre varios programadores que se tienen que comunicar entre ellos. Esto conduce al uso de diferentes niveles de abstracción que permiten derivar un programa ejecutable de una especificación descriptiva de la tarea original.
Estos conceptos de la programación, introducidos en los años 70 por Hoare y Dijkstra, serán aplicados y discutidos durante el curso en el contexto específico de la computación científica. Un aspecto nuevo del curso consiste en la modelización matemática de las tareas de un programa complejo (especificación descriptiva) en terminos de la teoría de bases de datos: la ejecución de tal programa será interpretada como la evaluación de una consulta. Este punto de vista permite por vez primera la demostración de cotas inferiores de complejidad relevantes para tareas no elementales de programación. Los ejemplos serán tomados de la aritmética y de la geometría.

Contenidos

Temas básicos:
Noción de algoritmo, programa y especificación. Diferentes modelos algoritmicos y medidas de complejidad. Ejemplos de estructuras de datos y algoritmos (in-) eficientes de la aritmetica y del álgebra lineal. Estructuras y tipos de datos. Longitud de programa y

tiempo de su ejecución. Introducción a la lógica de predicados y especificación formal. Bases de datos continuas y relacionales y sus lenguajes de consulta. La noción de consulta genérica, evaluación de consultas y su complejidad. Un modelo algoritmico adaptado al cálculo cientifico para la recursión en tiempo polinomial (extensión del modelo de Cook - Bellantoni).

Temas algorítmicos:

Complejidad aritmética y bit de algoritmos basicos simbolicos y numericos en algebra lineal con matrices genericos y estructuradas y su programación y verificación. El algoritmo de Newton y variantes simbolicos y numéricos. La estructura de datos straight-line program en programación orientada a objeto y programación funcional. Eliminación geométrica y algebraica (evaluación y reescritura). Analisis numérico y aproximación diofántica.

Especificación, verficación y derivación de programas:

Algunos tipos de datos fundamentales y su especificación algebraica. Especificación descriptiva de programas para tareas geométricos en terminos de la teoría de bases de datos contínuas. Ejecución de programas para tareas geométricas como evaluación de consultas. Transformación de programas descriptivos en operacionales mediante algoritmos recursivos. Diferentes tipos de recursión. Invariantes. . Cotas inferiores para el tiempo de ejecución de programas derivados de especificaciones en geometría.

12.- BIBLIOGRAFÍA (indicar título del libro, autor, Editorial y año de publicación)

J.L. Balcázar: Programación metódica, McGraw-Hill (1993)
J. L. Balcázar, D. L. Díaz, J. Gabarro: Structural complexity I, II, EATCS Mononographies on Theoret.. Comput. Sci. 11, Springer Verlag (1988)
P.Bürgisser, M. Clausen, M. A. Shokrollahi: Algebraic Complexity Theory, Springer Verlag (1997)
D. Castro, M. Giusti, J. Heintz, G. Matera, L. M. Pardo: The hardness of polynomial equation solving, Foundations of Computational Mathematics 3, 347-420 (2003)
J. Heintz, B. Kuijpers: Constraint data bases, data structures and efficient query elimination, aparecerá en Proc. Applications of Constraint Data Bases (CDB´04), Springer LNCS (2004)
G. M. Kuper, J. Paredaens, L. Libkin: Constraint data bases, Springer Verlag (1999)
B. Liskov, J. Guttag: Abstraction and Specification in Program Development, MIT Press (1986)
C. Morgan: Programming from Specifications, Prentice Hall (1990)
H. A. Partsch: Specification and transformation of programs, Springer Verlag (1990)
M Wirsing: Algebraic Specification, en Handbook of Theoretical Computer Science B, Elsevier (1990)

Dr. Alejandro N. Rios
Departamento de Computación
FCEyN UBA

UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
ENTRO | SALIO
13 FEB. 2006