2do cuatrimestre de 1991.

UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO: Computación.....

ASIGNATURA: Teoría de lenguajes y autómatas.....

CARRERA/S:..Licenciatura en Cs. de la Computación. (87).....
.....

CARACTER: OBLIGATORIA.....(indicar si es obligatoria u optativa)

PUNTAJE:.....(en caso de ser optativa)

DURACION DE LA MATERIA:.....(indicar si es cuatrimestral o anual).

HORAS DE CLASE: a) TEORICAS...3... HS. b) PROBLEMAS ...6... HS.
c) LABORATORIO... HS. d) SEMINARIOS..... HS.
e) TOTALES....9.. HS.

ASIGNATURAS CORRELATIVAS: PROGRAMACION DE COMPUTADORAS II....
.....

PROGRAMA:

1- Introducción

Las funciones fundamentales de un compilador. Estructura clásica, compilación pura, interpretación pura, soluciones intermedias. Lenguaje fuente, objeto de implementación, notaciones de Bratman y Rosin. Variantes producidas por la búsqueda de mayor interacción. Ambientes de programación. El problema de la especificación sintáctica y semántica del lenguaje. Reseña histórica y ejemplos.

2- Lenguajes formales

Cadenas sobre un alfabeto, lenguajes. Gramáticas, derivación, lenguaje generado. Clasificación de Chomsky, inclusión estricta de las clases, correspondencia con autómatas. Gramáticas independientes del contexto, BNF, reconocimiento, árbol de "parsing". Gramáticas ambiguas, eliminación de ambiguedad, lenguajes ambiguos. Gramáticas regulares, diagramas de transición, autómatas finitos no determinísticos (NAFs) y determinísticos (DAFs), equivalencias. Construcción de un autómata finito no determinístico (NFA) que acepte el lenguaje generado por una expresión regular - algoritmo de Thomson. Simulación de un NFA. Análisis de espacio/tiempo para cada solución. Construcción directa de un DFA a partir de una expresión regular. Autómatas pila determinísticos y no determinísticos. Gramáticas reducidas. Gramáticas regulares. Decidibilidad de algunos problemas sobre lenguajes y gramáticas.

3- Análisis lexicográfico

Conveniencia de su separación del análisis sintáctico. Uso de expresiones regulares y autómatas finitos. Reconocimiento de la subcadena más larga que satisfaga a la primera expresión de una lista de expresiones regulares. Generadores de analizadores lexicográficos, uso de LEX.

4- Análisis sintáctico

Construcción del Arbol de derivación de una sentencia, derivación más hacia la derecha y más hacia la izquierda. Analizadores descendentes. Descendencia recursiva. Eliminación de recursividad a izquierda.

CD 321192

Analizadores predictivos, conjuntos de símbolos selectores, algoritmo para calcularlos, condición para que un gramática sea LL(1), gramática LL(K). Análisis dirigido por tabla. Algoritmo de construcción de un analizador LL(1). Modificación de una gramática en busca de una LL(1) equivalente, factorización.

5- Métodos ascendentes.

Métodos de corrimiento y reducción, propiedad fundamental. Precedencia de operadores. Cálculo de relaciones y funciones de precedencia de operadores, algoritmo de construcción de analizadores de precedencia de operadores. Gramáticas LR(k), prefijos válidos, construcción de analizadores LR(0), SLR(1), LR(1) y LALR(1). Uso de gramáticas ambiguas. Tratamiento de errores en cada método.

6- Gramáticas de atributos.

Gramáticas de atributos, atributos heredados y sintetizados definiciones guiadas por sintaxis, esquemas de traducción. Evaluación de atributos, gráfico de dependencia, verificación de no circularidad. Gramáticas S atributadas y L atributadas evaluación durante el análisis descendente de gramáticas L atributadas, evaluación durante el análisis ascendente de gramáticas S atributadas, simulación de atributos heredados mediante el uso de símbolos demarcadores durante el análisis de corrimiento y reducción, posibilidad de evaluación de gramáticas LL(1) L atributadas, restricciones para las gramáticas LR(1). Evaluadores recursivos. Algunos casos no evaluables en una pasada.

7- Verificación de tipos.

Sistemas de tipos, tipos simples, constructores de tipos, expresiones de tipo. Equivalencia estructural y nominal. Coersión, sobrecarga de operadores y funciones. Verificación estática y dinámica.

8- Entorno de tiempo de ejecución.

Organización de la memoria. Registros de activación. Implementación del alcance de variables en tiempo de compilación y de ejecución, alcance estático y dinámico. Pasaje de parámetros, pasaje de procedimientos como parámetros. Alocación dinámica de memoria.

9- Código intermedio.

Representaciones usadas: árboles, DAGs, ternas, cuaternas, ternas indirectas, notación polaca generalizada. Representación de declaraciones, asignaciones, expresiones, optimización de expresiones lógicas, constructores de control, llamados a procedimientos. Esquemas de traducción correspondientes.

10- Generación de código.

Manejo de memoria en tiempo de ejecución. Bloques básicos y diagramas de flujo, alocación y asignación de registros, representación de bloques básicos mediante DAGs. Estudio de un generador de código.

Se realizarán trabajos prácticos usando YACC, LEX y módulos provistos por la cátedra.

Bibliografía:

Aho, Sethi, Ullman. Compilers, principles, techniques and tools. Addison Wesley, 1986.

Temblay, Sorenson. Compilers writing. Mc Graw Hill, 1986.

Barstow, Shrobe, Sandwall (Ed) Interactive Programming Environments.

Hunter. The design and construction of compilers. John Wiley, 1981.

Aho, Ullman. The theory of Parsing Translation and Compiling. Prentice Hall, 1973.

Pagan. Formal Specification of programming Languages. Prentice Hall, 1981.

Hopcroft, Ullman. Introduction to Automata Theory, Languages and Computation. Addison Wesley, 1979.

FECHA: 11/91.....

DOCENTE RESPONSABLE
JORGE AGUIRRE

AUTORIDAD DEPARTAMENTAL
BEVILACQUA