



ASIGNATURA: 12694 - COMPILADORES

CENTRO: Escuela de Ingeniería Informática

TITULACIÓN: Ingeniero en Informática

DEPARTAMENTO: INFORMÁTICA Y SISTEMAS

ÁREA: Lenguajes Y Sistemas Informáticos

PLAN: 10 - Año 199 **ESPECIALIDAD:**

CURSO: Cuarto curso **IMPARTIDA:** Segundo semestre **TIPO:** Troncal

CRÉDITOS: 4,5

TEÓRICOS: 3

PRÁCTICOS: 1,5

Descriptor B.O.E.

Compiladores. Traductores e intérpretes. Fase de compilación.

Temario

Nota: El número de horas se mantiene a efectos informativos para el estudiante para la estimación del peso de cada módulo dentro de la asignatura.

MÓDULO I: INTRODUCCIÓN [7 horas]:

([Pérez 98], [Aho 90], [Gries 75], [Sánchez 89], [Sanchís 88])

Tema 1. Introducción a los compiladores [2 horas].

Tema 2. Estructura de un compilador [2 horas].

Tema 3. 'Bootstrapping' de un compilador [2 horas].

Tema 4. Ejemplo sencillo de un compilador y de un intérprete [1 hora].

MÓDULO II: GRAMÁTICAS Y LENGUAJES [4 horas]:

([Pérez 98], [Aho 90], [Gries 75], [Alfonseca 87], [Sánchez 89], [Sanchís 88])

Tema 1. Definiciones de la teoría de lenguajes formales [0'5 horas].

Tema 2. Definición formal de gramática. Tipos de gramáticas [0,5 horas].

Tema 3. Ejercicios [1 hora].

Tema 4. Relaciones [0'5 horas].

Tema 5. Restricciones de una gramática [0'5 horas].

Tema 6. Notación BNF Extendida [0'5 horas].

Tema 7. Jerarquía de gramáticas de Chomsky [0'5 horas]

MÓDULO III: LÉXICO [4 horas]:

([Pérez 98], [Aho 90], [P4], [Gries 75], [Alfonseca 87], [LC99])

Tema 1. Autómatas finitos y expresiones regulares [2 horas].

Tema 2. Algunos ejemplos de analizadores léxicos [2 horas].

MÓDULO IV: SINTÁCTICO [15 horas]:

([Pérez 98], [Aho 90], [Wirth 80], [LC99], [Sánchez 89], [Sanchís 88], [Gries 75], [John-son 75], [Tremblay 85], [Sippu 90])

Tema 1. Análisis descendente [1 hora].

- Tema 2. Descenso Recursivo para BNF [1 hora].
- Tema 3. Condiciones en notación EBNF [1 hora].
- Tema 4. Traducción [1 hora].
- Tema 5. Compilador concreto: LC99(sintáctico) o PL/0 [1 hora].
- Tema 6. Reconocedores LL(1) [1 hora]
- Tema 7. Transformaciones de las gramáticas [1 hora].
- Tema 8. Análisis ascendente. Reconocedores LR [8 horas]

Requisitos Previos

Para esta asignatura es interesante haber cursado la de Autómatas, Gramáticas y Lenguajes Formales de 1º de II, así como tener soltura de programación en un lenguaje de alto nivel preferiblemente Pascal; y que el alumno esté familiarizado con diferentes lenguajes de programación tanto a nivel de usuario como de diseñador (estructuras, ámbito, etc.), para esto sería interesante que cursara como libre configuración cualquiera de las de Lenguajes de Programación de 2º de ITIS o ITIG.

Objetivos

El objetivo fundamental de dicha asignatura es que el alumno se familiarice con los compiladores y los intérpretes; tanto a nivel de usuario como de diseñador de los mismos. Conozca sus partes y en lo posible sea capaz de desarrollar la fase de análisis de sus propios compiladores e intérpretes. También es interesante el conocimiento de las técnicas que ello conlleva, para que las pueda utilizar en la construcción de sus propios diseños de interface con el usuario y/o lenguajes.

Metodología

El profesor expondrá los temas relacionados con la asignatura intentando sobre todo introducir la materia y orientar al estudiante, en consonancia con la disminución del número de horas presenciales que sufrirá la asignatura. En la presentación de la materia se emplearán los medios audiovisuales disponibles (pizarra, transparencias, videoproyectores, etc.). Se procurará que las clases sean lo más participativas posible, intentando motivar al alumno a que intervenga no sólo en la realización de ejercicios sino en la exposición de dudas e incluso en la resolución de las mismas. La asignatura seguirá manteniendo la necesaria realización práctica por parte del estudiante, preferiblemente en grupo, tras una breve introducción de la tarea a realizar (explicación de la práctica y resolución de dudas sobre la misma) y posteriormente comenzarán con el desarrollo de la misma.

Las tutorías serán un elemento básico de interacción profesor alumno, para resolver dudas y proporcionar a los alumnos interesados posibilidad de ampliar sus conocimientos sobre determinados temas, facilitándoles bibliografía y orientación.

Así mismo, se usará el Campus Virtual u otra herramienta similar para apoyar la docencia, tanto teórica como práctica y la tutorización de los alumnos; permitiendo acceso al material, entrega de trabajos prácticos, participación en foros, evaluaciones y test periódicos, etc.

Criterios de Evaluación

La nota final será al 50% la parte teórica y la práctica, siendo requisito aprobar ambas para superar la asignatura, truncándose a 4,5 si fuera necesario. Para la primera, se realizará un examen final, compuesto de igual número de partes que exámenes parciales se realicen durante el cuatrimestre, utilizándose para cada una de ellas, a efectos del cálculo de la nota de teoría, la mejor nota en cada una (parcial y final) de las partes. Respecto de la segunda, será la media aritmética de las distintas prácticas que se describen.

Descripción de las Prácticas

Práctica número 1: Definir formalmente un lenguaje de programación

Descripción

Cada grupo definirá su propio lenguaje formal e informalmente; para esta definición se podrán utilizar la notación EBNF o los Diagramas Sintácticos.

Objetivos

Esta práctica pretende que el alumno se habitúe con el uso de los metalenguajes, y que a su vez reflexione sobre los lenguajes de programación. El lenguaje diseñado por el alumno debe estar orientado a un tipo de problemas concretos, que se deben especificar en la documentación de la práctica.

Material de Laboratorio recomendado (Software)

Material de Laboratorio recomendado (Hardware)

No precisa material informático, puesto que el alumno la realizará en papel. Adicionalmente podría ser interesante que el alumno dispusiera de un PC con acceso a Internet para buscar un lenguaje candidato para su práctica.

Nº horas total estimadas para la realización de la práctica: 4 horas.

Práctica número 2: Analizador léxico construido manualmente

Descripción

En esta práctica se construye 'a mano' un analizador léxico del lenguaje que cada alumno tiene definido. No se ha definido todavía el lenguaje formalmente, ya que es en el módulo siguiente donde se estudia la notación EBNF y los diagramas sintácticos, pero cada alumno ya ha comunicado, de una manera informal, al profesor qué lenguaje ha elegido para su análisis por lo que sabemos qué símbolos terminales (tokens) ha de reconocer el analizador léxico y cual será la estructura de los programas escritos en dicho lenguaje. Básicamente, este programa lo que hará será aceptar un fichero de texto con un programa escrito en el lenguaje fuente e irá juntando los caracteres para formar los tokens que haya en el programa que se esté analizando.

Objetivos

Llevar a la práctica los conceptos sobre análisis léxico y desarrollarlos para un lenguaje concreto.

Material de Laboratorio recomendado (Software)

Sistema operativo WINDOWS-NT o LINUX, Turbo Pascal 7.0, DELPHI, GCC u otro compilador para el Lenguaje elegido por el grupo; algún procesador de texto tipo Word, alguna herramienta para representar diagramas de flujo, por ejemplo flow-char, smartdraw,etc..

Material de Laboratorio recomendado (Hardware)

Por cada 4 personas un PC que posea el software necesario para realizarla (sistema operativo WINDOWS-NT y LINUX, Turbo Pascal 7.0, DELPHI, GCC y los compiladores para el Lenguaje elegido; algún procesador de texto tipo Word, alguna herramienta para representar diagramas de flujo, por ejemplo flow-char, SmartDraw, etc.).

Nº horas total estimadas para la realización de la práctica: 8 horas

Práctica número 3: Construcción de un analizador sintáctico recursivo descendente

Descripción

Los alumnos, por un lado, tomarán la gramática diseñada en la práctica nº 1 y comprobarán si es aplicable el descenso recursivo, en caso negativo realizarán las transformaciones necesarias para que sí sea aplicable, sin modificar el lenguaje. Por otro lado, una vez transformada la gramática, construirá el analizador sintáctico correspondiente, ayudándose del 'Analizador léxico construido manualmente' desarrollado en la práctica nº 2.

Objetivos

Mentalizar al alumno de las condiciones que ha de tener la gramática de un lenguaje de programación, para realizar el análisis sintáctico descendente, de una manera determinista y sin retroceso. Familiarizar al alumno con una técnica de construcción de analizadores sintácticos descendentes.

Material de Laboratorio recomendado (Software)

Sistema operativo WINDOWS-NT o LINUX, Turbo Pascal 7.0, DELPHI, GCC u otro compilador para el Lenguaje elegido por el grupo; algún procesador de texto tipo Word, alguna herramienta para representar diagramas de flujo, por ejemplo flow-char, smartdraw, etc..

Material de Laboratorio recomendado (Hardware)

Por cada 4 personas un PC que posea el software necesario para realizarla (sistema operativo WINDOWS-NT y LINUX, Turbo Pascal 7.0, DELPHI, GCC y los compiladores para el Lenguaje elegido; algún procesador de texto tipo Word, alguna herramienta para representar diagramas de flujo, por ejemplo flow-char, SmartDraw, etc.).

Nº horas total estimadas para la realización de la práctica: 9 horas

Práctica de ampliación: A los trabajos anteriores se podrá añadir una tabla de símbolos, análisis semántico, generación de código, etc.

Bibliografía

[1 Básico] Traductores e intérpretes /

Miguel Angel Pérez Aguiar.

Universidad de Las Palmas de Gran Canaria, Escuela Universitaria de Informática,, Las Palmas de Gran Canaria :

(1998) - (2a ed.)

[2 Recomendado] Compiladores: principios, técnicas y herramientas /

Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman.

Addison-Wesley Iberoamericana,, Argentina : (1990)

0201629038

[3 Recomendado] Diseño de compiladores /

Alicia Garrido Alenda...[et al.].

Universidad de Alicante,, Alicante : (2004)

84-7908-700-5

[4 Recomendado] Compiladores: conceptos fundamentales /

Bernard Teufel, Stephanie Schmidt, Thomas Teufel.

Addison-Wesley Iberoamericana,, Argentina : (1995)

0201653656

[5 Recomendado] Construcción de compiladores /

David Gries.
Paraninfo,, Madrid : (1975)
8428305846

[6 Recomendado] Compiladores: teoría y construcción /

F. J. Sanchis LLorca, C. Galán Pascual.
Paraninfo,, Madrid : (1988) - (2ª ed.)
8428314691

[7 Recomendado] Compiladores e intérpretes: un enfoque pragmático /

Gonzalo Sánchez Dueñas, Juan A. Valverde Andreu.
Díaz de Santos,, Madrid : (1989) - (2ª ed.)
8487189067

[8 Recomendado] Construcción de compiladores: principios y práctica /

Kenneth C. Loudon.
Thomson,, Australia [etc.] : (2004)
970-686-299-4

[9 Recomendado] Lex & yacc.

Levine, John
O'Reilly and Associates,, Sebastopol, CA : (1995) - (2nd. ed.)
1565920007

[10 Recomendado] Teoría de lenguajes, gramáticas y automátas /

Manuel Alfonseca, Justo Sancho, Miguel [M]artínez Orga.
Universidad y Cultura,, Madrid : (1987)
8486367387

[11 Recomendado] Algoritmos + estructura de datos: programas /

Nicklaus Wirth ; versión castellana de Angel Alvarez Rodríguez, José Luis Cuenca Bartolomé.
..T260:
(1986)
8421901729

[12 Recomendado] Pascal implementation: compiler and assembler/interpreter /

Steven Pemberton and Martin Daniels.
Ellis Horwood,, New York : (1982)

[13 Recomendado] Pascal implementation: the P4 compiler /

Steven Pemberton and Martin Daniels.
Ellis Horwood,, New York : (1982)
0853125899

Equipo Docente

JOSÉ FORTES GÁLVEZ

(COORDINADOR)

Categoría: TITULAR DE UNIVERSIDAD

Departamento: INFORMÁTICA Y SISTEMAS

Teléfono: 928458724 **Correo Electrónico:** jose.fortes@ulpgc.es