

Ficha Técnica

Titulación:	Grado en Ingeniería Informática		
Plan BOE:	BOE número 108 de 6 de mayo de 2015		
Asignatura:	Autómatas y Procesadores de Lenguajes		
Módulo:	Lenguajes y Programación		
Curso:	3º	Créditos ECTS:	6
Tipo de asignatura:	Obligatoria	Tipo de formación:	Teórica-Práctica

Presentación

Procesadores de Lenguajes es una asignatura de tipo obligatorio que consta de 6 créditos y que introducirá al estudiante en las áreas fundamentales de la teoría de autómatas, lenguajes formales y gramáticas que constituyen la base del funcionamiento de los procesadores de lenguajes. Se exploran los problemas y técnicas que se plantean en la construcción de procesadores, compiladores, traductores e intérpretes de los diferentes lenguajes de programación. El estudio de dichas técnicas ofrecerá al estudiante una visión más amplia de los lenguajes de programación, previamente estudiados desde el punto de vista del programador y no del constructor de intérpretes o compiladores. Otras aplicaciones de la teoría de autómatas son los analizadores del lenguaje natural, los sistemas de control de máquinas industriales y la robótica.

En esta asignatura se abordarán principalmente los siguientes temas:

- Teoría de autómatas, que incluye autómatas finitos deterministas (AFD) y no deterministas (AFND), autómatas a pila.
- Lenguajes y expresiones regulares, y transformaciones.
- Gramáticas independientes del contexto.
- Análisis sintáctico y semántico.

Los requisitos previos obligatorios antes de abordar esta asignatura son:

- Es necesario contar con habilidades básicas de programación, que permitan desenvolverse en un entorno de Programación Orientada a Objetos, así como manipular estructuras de datos. Por lo tanto es obligatorio haber superado previamente la asignatura “Metodología de la Programación”.
- Es recomendable superar previamente la asignatura “Matemática Discreta”.

Competencias y/o resultados del aprendizaje

- Dominar los modelos formales que sostienen la teoría del procesamiento de lenguajes, tales como las expresiones regulares, teoría de autómatas y gramáticas.
- Describir las distintas fases y algoritmos utilizados en la traducción y generación de código desde el programa fuente al ejecutable, incluidas las consideraciones en la traducción de código dependiente e independiente de la máquina.
- Dominar los conceptos de optimización de código, incluyendo las distintas posibilidades en la elección de código intermedio y fases de optimización.
- Dominar los conceptos de tipos, ámbito y su comprobación (compatibilidad), en el procesamiento de

- lenguajes de programación.
- Ser capaz de utilizar herramientas de construcción de procesadores de lenguajes. Conocer el concepto de sistema de apoyo a la decisión y sus tipos.

Contenidos Didácticos

- 1 Introducción a los procesadores de lenguajes
 - 1.1 Concepto de traductor
 - 1.2 Historia de los procesadores de lenguajes
 - 1.3 Programas relacionados con los compiladores
 - 1.4 Proceso de compilación
- 2 Autómatas finitos deterministas
 - 2.1 Introducción a los autómatas finitos
 - 2.2 Autómata finito determinista (AFD)
 - 2.3 Diseño de AFDs
- 3 Autómatas finitos no deterministas
 - 3.1 Indeterminismo
 - 3.2 Definición de autómata finito no determinista (AFN)
 - 3.3 Lenguaje aceptado por un AFN
 - 3.4 Diseño de AFNs
- 4 Lenguajes y expresiones regulares
 - 4.1 Lenguajes regulares
 - 4.2 Expresiones regulares (ER)
 - 4.3 Diseño de ER
 - 4.4 Aplicaciones de las ER
- 5 Transformaciones entre modelos formales
 - 5.1 Transformación de ER en AFN
 - 5.2 Transformación de AFN a ER
 - 5.3 Algoritmo de construcción de subconjuntos
 - 5.4 Lema de bombeo
 - 5.5 Autómatas finitos con salida: Mealy y Moore.
- 6 Introducción a las gramáticas independientes del contexto
 - 6.1 Gramáticas
 - 6.2 Jerarquía de Chomsky
 - 6.3 Gramáticas regulares
 - 6.4 Gramáticas independientes del contexto (GIC)
 - 6.5 Diseño de GICs
- 7 Autómatas a pila
 - 7.1 Representación gráfica y simplificada
 - 7.2 Autómatas de pila deterministas
 - 7.3 Equivalencia entre GIC y autómatas de pila
 - 7.4 Árboles de derivación y derivaciones extremas
 - 7.5 Técnicas descendentes y ascendentes
 - 7.6 Forma normal de Chomsky
- 8 Análisis sintáctico descendente
 - 8.1 Introducción a las técnicas LL(1) y LL(k)
 - 8.2 Técnica LL(1)
 - 8.3 Método recursivo-descendente
 - 8.4 Tabla de análisis sintáctico descendente
 - 8.5 Transformaciones en las gramáticas
- 9 Análisis sintáctico ascendente

- 9.1 Introducción al análisis sintáctico ascendente
- 9.2 Técnicas LR
- 9.3 Técnica SLR(1)
- 9.4 Técnica LR(1)
- 9.5 Técnica LARL(1)
- 9.6 Proceso general LR
- 10 Análisis semántico
 - 10.1 Semántica de los lenguajes de programación
 - 10.2 Gramáticas con atributos
 - 10.3 Relación entre gramáticas con atributos y procesadores de lenguajes
 - 10.4 Gramáticas y lenguajes de marcado
 - 10.5 Transformación entre GIC con expresiones regulares a gramáticas independientes del contexto ordinarias
 - 10.6 Conclusiones

Contenidos Prácticos

Durante el desarrollo de la asignatura se realizarán las siguientes actividades prácticas:

- Resolución de casos prácticos de AFD y AFN.
- Resolución de casos prácticos de Transformaciones formales.
- Resolución de casos prácticos de Análisis sintáctico.

Evaluación

El sistema de evaluación del aprendizaje de la UDIMA contempla la realización de diferentes tipos de actividades de evaluación y aprendizaje. El criterio de valoración establecido se detalla a continuación:

Actividades de aprendizaje	10%
Controles	10%
Actividades de Evaluación Continua (AEC)	25%
Examen final presencial	55%
TOTAL	100%

Bibliografía

- Lezcano Matías, L. y Sánchez Alonso, S. Procesadores de Lenguajes, Ed. CEF.