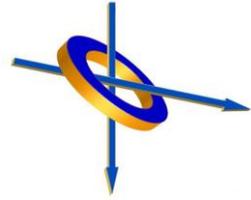




**UNIVERSIDAD DE COSTA RICA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS DE LA
COMPUTACIÓN E INFORMÁTICA**



**Curso: CI-2700
Grupo: 01**

**Tópicos Especiales - Compiladores
Segundo Ciclo Lectivo del 2014**

1. Presentación

Profesor:

Dr. Manuel Enrique Bermúdez Aguilar (manuel.bermudez@ecci.ucr.ac.cr), Oficina 238 IF.

Sitio Web del Profesor: <http://www.cise.ufl.edu/~manuel>

Sitio Web del Curso: <http://www.cise.ufl.edu/~manuel/UCR/curso-compiladores/>

Créditos: 4.0

Requisitos: CI-1221 Estructuras de Datos y Análisis de Algoritmos

Modalidad: Regular

Nivel: III año

Horario: Lunes y Jueves, 11:00 – 12:50 hrs, aula: 203 IF

Consulta: Lunes y Jueves 13:30 – 15:00 hrs.

2. Objetivos Generales

- Guiar al estudiante en el aprendizaje de técnicas de traducción de lenguajes de programación.
- Guiar al estudiante en el aprendizaje de técnicas de análisis léxico y análisis sintáctico, análisis de semántica estática y generación de código traducido, así como los principios teóricos que los fundamentan.

3. Objetivos Específicos

Al finalizar este curso el estudiante será capaz de:

- Implementar un compilador sencillo para un lenguaje de programación.
- Reconocer la diferencia entre un lenguaje regular y uno libre de contexto, e implementar algoritmos que reconocen estos tipos de lenguajes.
- Utilizar las herramientas necesarias para automatizar el proceso de generación de analizadores léxicos y sintácticos, para lenguajes de programación.
- Extender la implementación de un lenguaje de programación, agregando operadores, sentencias, tipos de datos nuevos, y funciones o procedimientos.

4. Unidades Temáticas y Contenidos

- A. Paradigmas y el Aprendizaje.
- B. Resumen de Compilación.

- C. Bootstrapping.
- D. El sistema TWS.
- E. Discusión Proyecto 0 (instalar el TWS).
- F. Precedencia de Operadores.
- G. Discusión Proyecto 1 (implementar operadores).
- H. Teoría de Lenguajes.
 - i. Definiciones.
 - ii. Ejemplos.
 - iii. Especificación de lenguajes.
 - iv. Constructores de Lenguajes.
 - v. Gramáticas.
 - vi. Jerarquía de Chomsky.
- I. Discusión Proyecto 2 (agregar sentencias).
- J. Lenguajes Regulares.
 - i. Gramáticas Regulares
 - ii. Regularidad Izquierda y Derecha.
 - iii. Conversión entre Gramáticas y Autómatas finitos.
 - iv. No-Determinismo.
- K. Discusión Proyecto 3 (agregar tipos de datos).
- L. Expresiones Regulares.
 - i. Conversión de gramáticas a expresiones regulares.
 - ii. Conversión de expresiones regulares a autómatas finitos.
 - iii. Eliminación de no-determinismo.
 - iv. Minimización de estados de un autómata finito.
- M. Lenguajes Libres de Contexto.
 - i. Árboles de derivación.
 - ii. Ambigüedad.
 - iii. Reducción de una gramática.
 - iv. Precedencia y Asociatividad.
 - v. Gramáticas de transducción.
 - vi. Árboles de Sintaxis Abstracta.
 - vii. El juego de Dominó Sintáctico.
 - viii. Estrategias de Análisis Sintáctico.
 - ix. Análisis Sintáctico Descendente.
- N. Discusión Proyecto 4 (agregar funciones)
- O. Análisis Sintáctico LL.
 - i. Conjuntos First, Follow, Select.
 - ii. Gramáticas LL(1).
 - iii. Transducción hilera-árbol.
 - iv. Generación descendente/ascendente del árbol de derivación.
 - v. Reemplazo de recursividad con iteración.
 - vi. Generación (ascendente) del árbol de derivación para la gramática original.
 - vii. Generación (ascendente) del árbol de sintaxis abstracta para la gramática original.
- P. Análisis Sintáctico LR.
 - i. El Autómata LR(0).
 - ii. Conflictos y su resolución: SLR(1), LALR(1).
- Q. Resumen de Análisis Sintáctico.

5. Evaluación

Aspecto evaluativo	Porcentaje
Exámenes parciales (20% x 2)	40%
Examen Final	20%
Proyectos (8% x 5)	40%

6. Metodología

En este curso se recomienda que el estudiante se prepare para cada clase mediante la lectura previa del material a cubrir. Habrá 2 **exámenes parciales y un examen final**. Las fechas de los exámenes serán anunciadas con un mínimo de una semana de anticipación. Se estima que el **primer examen parcial será a finales de setiembre, y el segundo a finales de octubre**.

Los proyectos serán individuales, a menos que se indique lo contrario en el enunciado del proyecto. El **proyecto cero** consistirá en montar el Sistema de Escritura de Traductores (TWS) en la plataforma Linux. Los proyectos **uno al cuatro** consistirán de agregar construcciones al lenguaje de programación: operadores, sentencias, tipos de datos, y funciones.

No se aceptarán trabajos después de la fecha estipulada de entrega y sólo se repondrán evaluaciones bajo el procedimiento descrito en el artículo 24 de Reglamento de Régimen Académico Estudiantil de la UCR.

El plagio no será tolerado en ninguno de los trabajos del curso (exámenes y proyectos) y será castigado con una nota de cero en el trabajo correspondiente, aún cuando el plagio se haya dado en sólo una parte del mismo. El profesor podrá además elevar el caso de plagio a las instancias respectivas de la universidad.

NOTA: Este documento es la versión inicial de las políticas del curso. La versión actual del contenido del curso, así como el material, políticas, fechas y anuncios, se encuentra en todo momento en la página Web del curso. Todo material, políticas, y fechas están sujetas a cambios por parte del profesor, siempre que el cambio se haga con una antelación adecuada y razonable. El estudiante es responsable de entrar a la página Web del curso (particularmente la página de Anuncios) en forma diaria, y de estar al día con el progreso del curso.

7. Bibliografía

Libros de Texto

1. Apuntes (Powerpoint) proveídos por el profesor en el sitio Web del curso.
2. Aho, Sethi, Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1986.

8. Cronograma

Las 16 Unidades Temáticas en la Sección 4 corresponderán a una semana de clases cada una, aproximadamente (algunas más, otras menos).