Programa del curso

Semestre 2020-20

Nombre del curso:	ISIS1106 Lenguajes y máquinas
Course Name:	------
Créditos:	3
Profesor:	---
Versión PDF	Click Aquí

Descripción

El objetivo de este curso es enseñar los conceptos básicos de la teoría de lenguajes a través de los formalismos usados para describir los lenguajes y las máquinas usadas para reconocerlos. Se pretende estudiar el uso de estas máquinas en dominios distintos a los lenguajes. Finalmente se verán otras máquinas abstractas y su utilidad en modelaje.

Generalidades

- Clases: 3 horas semanales (dos sesiones de 1:30), de asistencia obligatoria. Durante las clases el profesor llevará una bitácora de presencia de los estudiantes como registro de asistencia. El estudiante que no asista al menos al 80% de las clases y sesiones de trabajo supervisado no podrá aprobar el curso, de acuerdo con el artículo 42 y 43 del RGRPr.
- Todos los trabajos y las pruebas serán individuales.
- El curso tiene como canales oficiales de comunicación el correo electrónico uniandes, la lista de correo del curso y el sistema de apoyo a la docencia SICUA+ (http://sicuaplus.uniandes.edu.co).

Objetivos

Al final del curso el estudiante debe estar en capacidad de:

- Entender los distintos formalismos usados para definir lenguajes.
- Modelar problemas sencillos con distintos tipos de máquinas abstractas.
- Entender las limitaciones de cada una de éstas.
- Usar lenguajes formales para definir las máquinas.
- Demostrar propiedades sobre estas máquinas.

Objetivos pedagógicos transversales

Objetivo Pedagógico	Metas específicas

OT1	Integrar módulos que apliquen técnicas de máquinas abstractas / lenguajes a aplicaciones existentes
OT6	Mejorar herramientas existentes aplicando estas técnicas
OT8	Autoaprender desarrollando

Categorias de habilidades y objetivos pedagógicos

Objetivo pedagógico	Metas específicas	%
O2	Entender el contexto donde es aplicable el análisis de lenguajes y el modelaje con máquinas abstractas	12%
O3	Saber cómo integrar el modelaje con máquinas y el análisis de lenguajes a problemas computacionales	12%
O4	Entender los formalismos para definir lenguajes, máquinas abstractas y su comportamiento.	24%
O5	Ser capaz de saber dónde integrar los lenguajes y las máquinas a un proyecto de SW	12%
O9	Diseñar máquinas abstractas para resolver problemas.	12%
O10	Emplear diagramas y simuladores para validar los diseños realizados	6%
O11	Diseñar la integración de las máquinas en proyectos más complejos usando los formalismos de descripción de lenguajes.	12%
O13	Implementar máquinas abstractas a partir de especificaciones formales. Usar herramientas existentes para ayudar en el análisis de lenguajes.	11%

Evaluación

1. Exams: 45%

- Final Exam: 30%
- Quizzes: 15%

2. Participation 5%
3. PQs: Practical quizzes:20%
4. Projects: 30%

- Project 0 6%
- Project 1: JavaCC: 7%
- Project 2: Mealy/Moore automaton: 6%
- Project Pharo: (deterministic automaton): 4%
- Project 3: Lexer and Parser with GOLD 7%

Notas Finales

Política de aproximación de notas finales Las notas definitivas del curso varían entre 1.50 a 5.00, en intervalos de 0.5. La asignación de la nota se determinará teniendo en cuenta el desempeño de todo el curso. Se ordenarán y agruparán estudiantes según la nota final y a todos los estudiantes de un mismo grupo se les asignará la misma nota.

Grupo 1 5.0
Grupo 2 4.5
Grupo 3 4.0
Grupo 4 3.5
Grupo 5 3.0
Grupo 6 2.5
Grupo 7 2.0
Grupo 8 1.5

Asistencia a clase: La asistencia a clase es obligatoria. Los estudiantes deben asistir a por lo menos el 80% de las clases del semestre. **Colaboración y trabajo en grupo:** Todos los trabajos y las pruebas serán individuales a no ser que se indique lo contrario en el enunciado. **Revisión de entregas:** Todas las entregas serán revisadas mediante procesos manuales y automáticos para detectar similitudes. [tab title="Metodología"] Presentaciones de la teoría y talleres. Trabajo en clase resolviendo problemas.

Metodología

Presentaciones de la teoría y talleres. Trabajo en clase resolviendo problemas. [tab title="Recursos"]

Bibliografía

Notas de Clase

Capítulo 1 Capítulo 2 Capítulo 3 Capítulo 4 Capítulo 5 - (Versión Preliminar) REDES DE PETRI

Presentaciones

Sintaxis: presentación incial Expresiones Regulares Ejemplo BNF extendido Otro ejemplo BNF Presentación JavaCC

Textos usados en el curso

1. Takahashi, S. Notas del Curso: Se irán publicando en la WIKI del curso
2. Manuales de JavaCC https://javacc.dev.java.net/
3. Harry R. Lewis and Christos H. Papadimitriou. 1997. Elements of the Theory of

Computation (2nd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.
4. Petri nets: http://en.wikipedia.org/wiki/Petri_net

Textos de referencia

1. Aho, A.V., Lam, M., Sethi, R., Ullman J.D. Compilers Principles, Techniques, & Tools, 2nd Edition, Addison Wesley, 2007.

1. Hopcroft, J.E., Motwani, R., Ullman, J.D., Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Addison Wesley, 2007.

Ejemplos

eclipse JavaCC

Ejemplos JavaCC

ParserTester Ejemplo solución al taller 2012-10 Gramática de Hechos y Reglas Prolog:archivo jj

Ejemplos Autómatas de estados finitos

Probador de Autómatas Ejemplos para el PROBADOR Messenger_Amigos

Ejemplos Autómatas de pila

Sitio del probador de autómatas de pila: Ejemplos y jar del probador de autómatas de pila presentacion1.pdf ejemplos.zip palindromes

Ejemplos máquinas de Turing

sumar10 tm_examples.zip ejemplo.zip= Ver Sicua+

Redes de Petri

presentaciones Simulador: Enlace externo (Version que se está usando) Nueva Version: Enlace externo [tab title="tutoriales"]

Tutorial Autómata Tester

Tutorial AutomataTester Ejemplos (solucion taller 201110)

Tutorial MessengerAmigos

Tutorial MessengerAmigos □□- CON GOLD □□-

1. 1. Se implementan los autómatas con respuesta en un proyecto GOLD (sugiero usar ISIS1106_Ejemplos_v3).
1. 2. Se copian los archivos .java correspondientes, ubicados en el directorio \ISIS1106_Ejemplos_v3\src-gen\uniandes\cupi2\messengerAmigos\verificador\gold

1. 3. Se pegan esos archivos en el directorio sources\uniandes\cupi2\messengerAmigos\verificador\gold del proyecto Messenger Amigos, abierto en cualquier versión de Eclipse.

Tutorial ParserTester

Tutorial ParserTester

Tutorial Redes de Petri

Tutorial Redes de Petri

Tutorial PDA Simulator

Tutorial simulador autómatas de pila

Tutorial VAS

Tutorial VAS

Tutorial Gold

Manuales

Documento de tesis de GOLD: gold3_documento.pdf Manual de instalación de GOLD: gold3_manualinstalacion.pdf Manual de solución de problemas de GOLD: gold3_manualproblemas.pdf

Instaladores(28 Nov del 2012)

Eclipse/Xtext con GOLD 3.1.3 - Windows 32 bits: eclipse_gold-3.1.3_win32.zip
Eclipse/Xtext con GOLD 3.1.3 - Windows 64 bits: eclipse_gold-3.1.3_win64.zip
Eclipse/Xtext con GOLD 3.1.3 - Mac OS X Cocoa 32 bits: eclipse_gold-3.1.3_mac32.zip
Eclipse/Xtext con GOLD 3.1.3 - Mac OS X Cocoa 64 bits: eclipse_gold-3.1.3_mac64.zip
Eclipse/Xtext con GOLD 3.1.3 - Linux GTK 32 bits: eclipse_gold-3.1.3_linux32.zip

Eclipse/Xtext con GOLD 3.1.3 - Linux GTK 64 bits: eclipse_gold-3.1.3_linux64.zip Todos los instaladores se encuentran en el siguiente enlace: Aqui

Recursos adicionales

JFLAP modificado para aceptar cadenas con estado final y pila vacía: jflap.zip Ejemplos varios (versión v3): isis1106_ejemplos_v3.zip Proyecto MessengerAmigos con la librería de GOLD: messengeramigosgold.zip GOLD Turing Machine Editor/Simulator (v3.1.3: 28/11/2012, 18:12h) gold-tm-3.1.3.zip Ejemplos: **Ítem de lista desordenada** **- Instrucciones para usar MessengerAmigos** **-** 1. Se implementan los autómatas con respuesta en un proyecto GOLD (se sugiere usar ISIS1106_Ejemplos_v3). 2. Se mueven los archivos .gold al directorio src\uniandes\cupi2\messengerAmigos\verificador\gold del proyecto GOLD. 3. Se copian los archivos .java correspondientes, ubicados en el directorio src-gen\uniandes\cupi2\messengerAmigos\verificador\gold. 4. Se pegan esos archivos en el directorio source\uniandes\cupi2\messengerAmigos\verificador\gold del proyecto Messenger Amigos, abierto en cualquier otra versión de Eclipse. 5. Se siguen las instrucciones consignadas en Tutorial MessengerAmigos: 5.a. En la clase VerificadorMensajes ubicada en el directorio source\uniandes\cupi2\messengerAmigos\verificador, se añade una nueva entrada en el arrego checks para cada nuevo autómata con respuesta, como se hace justo debajo del comentario /* extensiones GOLD */. 5.b. En la interfaz IVerificadorMensajes ubicada en el mismo directorio se añade una nueva entrada en el arreglo names para cada nuevo autómata con respuesta registrado en el paso anterior, respetando el orden de los elementos.