



INSTITUTO/S: Tecnología e Ingeniería

CARRERA/S: Licenciatura en Informática

MATERIA: Lenguajes Formales y Autómatas

NOMBRE DEL RESPONSABLE DE LA ASIGNATURA: Mg. Pablo Pandolfo

EQUIPO DOCENTE: -----

CUATRIMESTRE: 2^{do}

AÑO: 4^{to}

PROGRAMA N°: 30

(Aprob. Por Cons.Directivo 04/06/2022)

Instituto/s: Tecnología e Ingeniería

Carrera/s: Licenciatura en Informática

Nombre de la materia: Lenguajes Formales y Automatas

Responsable de la asignatura y equipo docente: Mg. Pablo Pandolfo

Cuatrimestre y año: Segundo cuatrimestre del cuarto año.

Carga horaria semanal: 4 hs

Programa N°: 30

Código de la materia en SIU: 775

Lenguajes Formales y Automatas

1. Fundamentación

Esta materia presenta al alumno/a los fundamentos teóricos formales para el entendimiento del funcionamiento de los compiladores y lenguajes de programación. La misma se centra en la tarea de reconocimiento de los elementos del lenguaje y su análisis sintáctico.

2. Propósitos y/u objetivos

Propósitos

- Promover la reflexión sobre el estudio de los procesos computacionales y explorar el alcance de los mismos en el contexto de una jerarquía de autómatas.
- Brindar las bases de la teoría de autómatas (autómatas finitos, autómatas de pila y máquina de Turing) y lenguajes formales (de acuerdo a la jerarquía de Noam Chomsky).
- Contribuir al análisis del poder computacional de los autómatas en el contexto de la resolución de problemas de reconocimiento de patrones.
- Experimentar de forma gráfica con los conceptos relacionados a la teoría de autómatas y lenguajes formales, permitiendo diseñar, evaluar y realizar distintas transformaciones y comprobaciones sobre autómatas finitos, gramáticas, autómatas a pila y máquinas de Turing.

Objetivos

Que el /la alumno/a sea capaz de:

- Reconocer los fundamentos teóricos de los lenguajes formales.
- Desarrollar la teoría de lenguajes formales a partir de las operaciones con cadenas.

- Enunciar las operaciones entre lenguajes formales en cuanto a conjunto de cadenas.
- Describir analíticamente las diferencias y relaciones entre gramáticas, lenguajes formales y autómatas.
- Demostrar el valor algorítmico de autómatas finitos y de pila y a qué problemas pueden ser aplicados.
- Reconocer la relación entre la teoría de las matemáticas, la lingüística y la informática.

3. Programa sintético:

Lenguajes y gramáticas. Gramáticas e isomorfismos. Jerarquía de Chomsky. Lenguajes regulares. Autómatas. Expresiones regulares. Minimización de autómatas. Analizadores lexicográficos. Lenguajes independientes de contexto. Árboles de derivación. Autómatas de pila. Lenguajes determinísticos. Lenguajes tipo 1 y tipo 0. Máquinas asociadas. Máquinas de Turing. Problema de la detención.

4. Programa analítico

4.1 Organización del contenido:

Unidad 1: Lenguajes Formales. Definición de Lenguaje Formal, Símbolo, Alfabeto y Palabra. Palabra: longitud de palabra, palabra vacía, concatenación de dos palabras, potenciación de una palabra, igualdad de palabras, reflexión o reversa de una palabra, prefijo de una palabra, sufijo de una palabra y subpalabra de una palabra. Símbolo: potenciación de un símbolo. Lenguaje Formal: diferencias con el lenguaje natural, cardinalidad de un lenguaje formal, lenguajes formales finitos e infinitos, sublenguajes de un lenguaje formal, lenguaje universal, lenguaje sobre un alfabeto, operaciones con lenguajes: unión, intersección, complemento, diferencia, producto, potencia, cierre estrella o de Kleene, cierre positivo y reflexión. Propiedades de los lenguajes.

Unidad 2: Gramáticas Formales. Definición de Gramática Formal y Producción. Definición formal. Sistemas de reescritura. Jerarquía de Chomsky. Tipos de gramáticas: tipo 3 o gramática regular, tipo 2 o gramática independiente de contexto, tipo 1 o gramática dependiente de contexto y tipo 0 o gramática irrestricta. Restricciones en las producciones. Concepto de derivación. Tipos de derivación: derivación en forma horizontal y derivación en forma vertical. Árboles de derivación. Forma sentencial. Sentencia. Lenguaje generado por una gramática. Gramáticas equivalentes. Gramáticas ambiguas y no ambiguas. Recursividad. Niveles de recursividad: producción recursiva, gramática recursiva, recursividad por la izquierda y por la derecha. Factorización a izquierda. Eficiencia en el diseño de gramáticas: limpias y bien formadas. Formas Normales: Forma Normal de Chomsky (CNF), Forma Normal de Greibach (GNF) y Forma Normal de Backus-Naur (BNF).

Unidad 3: Lenguajes Regulares. Definición de Lenguaje Regular. Ejemplos. Gramática de un lenguaje regular. Expresiones regulares. Componentes. Precedencia de operadores. Lenguaje representado por una expresión regular. Propiedades de equivalencias entre expresiones regulares. Expresiones regulares equivalentes y no equivalentes. Expresión Regular Universal. Expresiones regulares y los lenguajes de programación. Autómata finito. Concepto. Definición formal. Representación gráfica a través de grafos. Representación tabular a través de la Tabla de Transiciones. Reconocimiento de palabras. Diseños incompletos e incorrectos. Autómatas Finitos Incompletos y Completos. Autómatas Finitos equivalentes. Autómatas Finitos Determinísticos. Autómata Finito No Determinísticos. Operaciones con autómatas finitos: complemento, intersección, unión y concatenación. Algoritmos con Autómatas Finitos: (ER \rightarrow AFN λ), (AFN λ \rightarrow AFD), (AFD \rightarrow AFDmin), (AF \rightarrow ER), (GR \rightarrow AFD), (AFD \rightarrow GR). Implementación de autómatas finitos. Lema de pumping para lenguajes regulares. Aplicaciones.

Unidad 4: Lenguajes Incontextuales. Definición de Lenguaje Independiente de Contexto. Gramática de un lenguaje independiente de contexto. Autómata de Pila. Definición formal. Diseño con transiciones especiales. Autómatas de Pila Deterministas. Autómatas de Pila No Deterministas. Lenguaje aceptado por un Autómata de Pila. Reconocimiento de palabras por estado final y por pila vacía. Análisis de palabras. Lema de pumping para lenguajes independientes de contexto. Análisis sintáctico descendente (ASD): Analizadores sintácticos descendentes con retroceso. Analizadores sintácticos descendentes predictivos. Conjuntos de predicción y gramática LL(1). Conversión a gramática LL(1). Tabla de análisis sintáctico LL. Análisis sintáctico ascendente (ASA): Algoritmo de desplazamiento y reducción. Analizadores sintácticos ascendente con retroceso. Analizadores sintácticos ascendentes predictivos (SLR).

Unidad 5: Máquina de Turing. Definición de Máquina de Turing. Operación de la máquina de Turing. Diferencias de la máquina de Turing con un autómata finito. Definición formal de máquina de Turing. Configuración de una máquina de Turing. Lenguaje reconocido por una Máquina de Turing. (Turing-acceptable). Aceptación y rechazo de palabras por Máquinas de Turing. Problema de la detención o parada.

4.2 Bibliografía y recursos obligatorios:

Hopcroft, J.; Motwani, R. y Ullman, J. (2007). *Teoría de autómatas, lenguajes y computación.* España, Madrid: Editorial Addison Wesley.

4.3 Bibliografía optativa:

Cubero, E.; Moreno, M. y Salomón R. (2007). *Teoría de autómatas y lenguajes formales.* España, Madrid: Editorial McGraw-Hill.

Kelley, D.. (1995). *Teoría de Autómatas y Lenguajes Formales.* UK, Editorial Prentice Hall.

Isasi, Martinez y Borrajo. (1997). *Lenguajes, Gramáticas y Autómatas.* España, Madrid: Editorial Addison Wesley.

5. Metodologías de enseñanza:

Las clases serán divididas en contenido teórico y práctico.

Por motivos prácticos ambas clases se dictan en el mismo día y horario; y los segmentos prácticos y teóricos se estructuran en función de los temas dictados.

1. Clases de teoría. Durante las clases teóricas se presentarán los contenidos teóricos, se demostrarán los teoremas y propiedades de cada uno de los conceptos integrados y se resolverán ejemplos similares a los de la práctica que permiten la visualización de los conceptos y técnicas necesarias para la resolución de los mismos.
2. Clases prácticas. En las clases prácticas se realizarán ejercicios similares a los encontrados en la guía de trabajos prácticos y los que serán evaluados en los parciales. Se atenderán todas las dudas de los/as alumnos/as y se explicarán todos aquellos temas necesarios para la implementación de los algoritmos y técnicas en un lenguaje de programación. En todas las clases prácticas los/as alumnos/as sabrán de antemano qué problemas tienen que realizar. En ellas el/la profesor/a dará indicaciones para que los/as alumnos/as puedan superar las dificultades que les hayan aparecido en su resolución y resolverá individualmente las dudas y problemas que les hayan aparecido durante su resolución. Será labor del grupo de alumnos/as elaborar la colección de problemas resueltos de la asignatura.
3. Descripción de las Actividades Prácticas. Resolución de Problemas del Mundo Real: Los/as alumnos/as deberán resolver Trabajos Prácticos que contienen problemáticas referidas a cada uno de los núcleos temáticos vistos, que corresponden a cada una de las Unidades de los contenidos analíticos. No son de entrega obligatoria. Uso de la herramienta JFlap para modelar cada una de las máquinas abstractas vistas.
4. Simulacros de parcial. La clase previa a cada evaluación parcial se llevará un simulacro que consistirá en plantear a los/as alumnos/as situaciones problemáticas similares a las que pueden evaluarse. Instándolos a resolverlos con el apoyo del material elaborado durante las clases, y luego haciendo una puesta en común con el docente y demás alumnos/as, lo que les permitirá hacer una valoración crítica ante la instancia parcial.

Plan de trabajo en el campus:

El aula virtual de la materia se concibe como un espacio de extensión de la clase presencial. Por lo tanto, se subirá todo el material educativo como: apuntes de clase, videos introductorios o ampliatorios de los temas de clase, ejercicios resueltos, guías de trabajos prácticos y los Trabajos prácticos N°1 y N°2 para la entrega de los mismos. También incluirá un foro de consultas, el programa y el cronograma de la asignatura.

6. Actividades de investigación y extensión (si hubiera)

No aplica.

7. Evaluación y régimen de aprobación

7.1 Aprobación de la cursada

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

El/La alumno/a deberá poseer una asistencia no inferior al 75% en las clases presenciales.

En cuanto a la cursada de manera virtual se requerirá que el/la estudiante ingrese al aula virtual como mínimo una vez por semana.

7.2 Aprobación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

Evaluación integradora: tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

7.3 Criterios de calificación

Los/as alumnos/as deberán conocer los principios de los lenguajes formales regulares e independientes del contexto, las gramáticas que generan esos lenguajes y dominar los conceptos de autómatas necesarios para su identificación; además deben ser capaces de reconocer propiedades y realizar pruebas sobre las mismas a través de demostraciones aritméticas. Comprendiendo las capacidades y limitantes de cada una de estas herramientas y las mejores alternativas para su implementación.

Para demostrar estos conocimientos deben demostrar suficiencia en la resolución de problemas teóricos prácticos, demostraciones e implementaciones en un lenguaje de programación.

Los problemas teóricos prácticos presentados se van correspondiendo con los contenidos dictados en la materia y se trabajan durante la misma, para presentar las técnicas necesarias para resolver dichos problemas.

8. Cronograma

El siguiente cronograma se establece en función de los temas de las diferentes unidades del contenido de la materia. Las clases se conformarán de manera presencial, virtual o combinadas y se comunicarán al inicio de la cursada.

Semana	Unidad (teoría y ejercicios)
1	Presentación de la materia. Lenguajes Formales. Clasificación de Chomsky.
2	Gramáticas Formales. Gramáticas Regulares. Expresiones Regulares.
3	Autómatas Finitos.
4	Algoritmos con Autómatas Finitos.
5	Lema de pumping para LR. Entrega TP 1.
6	Repaso general. Resolución de Ejercicios.
7	Primer parcial.
8	Lenguajes Incontextuales. Gramáticas Independientes de Contexto. Formas Normales (FNC, FNG y FNB).
9	Autómatas a Pila.
10	Lema de pumping para LIC.

11	Parsers LL(k) y LR(k) con retroceso. Gramáticas LL(1) y Análisis Predictivo
12	Resolución de ejercicios.
13	Máquina de Turing. Máquina de Turing con variaciones. Entrega TP 2.
14	Repaso general. Resolución de Ejercicios.
15	Segundo parcial.
16	Recuperatorio y Recuperatorio TP. Cierre de Cursada.