

INICIATIVA ACADÉMICA DE TEORÍA DE LA COMPUTACIÓN

1 Identificación

Curso:	CC2019 – Teoría de la Computación	Créditos:	5
Ciclo:	Segundo	Requisitos:	Estructuras de datos y Algoritmos Matemática Discreta 1 Lógica Matemática
Año:	2022		
Profesor:	Alan Reyes-Figueroa	Horario:	Lunes – 19:50-21:25 – CIT 315B
Email:	agreyes	Sala:	Miércoles – 19:00-21:25 – CIT 301

Sitio Web del Curso:

- <https://pfafner.github.io/tc2022>

Office Hours:

- Viernes de 19:00 a 20:00 hrs, o por solicitud del estudiante. También pueden enviar sus dudas a través del correo electrónico institucional.

2 Descripción

Este es un curso introductorio a la teoría de la computación, la cual se ocupa de determinar qué problemas pueden ser resueltos computacionalmente y con qué eficiencia, así como de entender el límite entre los problemas computables y los no-computables, y clasificarlos de acuerdo a su simpleza o dificultad. El curso inicia con el estudio de distintos modelos de cómputo, como los autómatas finitos (que son los más sencillos), y sus diferentes tipos y aplicaciones; las máquinas de Turing (que son las computadoras usuales de hoy en día) y las computadoras cuánticas (cuyo funcionamiento no es digital).

Una vez formulado un modelo de computación, nos interesa conocer la cantidad de recursos computacionales que es necesario utilizar para resolver un problema. El primero de estos recursos es el tiempo: cuántos pasos o cuantas acciones debemos realizar para resolver el problema. También son importantes el espacio ocupado, la necesidad de utilizar una fuente de números aleatorios, la posibilidad de resolver subproblemas en paralelo, entre otros.

La formalización de un modelo de computación como objeto matemático permite expresar de manera precisa preguntas sobre problemas o algoritmos. En particular, si L es un problema. ¿Puede L ser resuelto por un algoritmo? ¿Puede ser L resuelto de manera eficiente? ¿Cómo podemos construir un algoritmo eficiente para L ? ¿Es L un problema para el cual no existe un algoritmo que lo resuelva? Contestaremos a algunas de estas preguntas, mientras que otras se verán en los cursos de Lógica Matemática y en Análisis de Algoritmos.

El curso cuenta con una parte práctica, en la que el estudiante implementará en código computacional algunos de los modelos estudiados. Parte fundamental del curso es utilizar las herramientas aprendidas en varios proyectos aplicados donde se utilizan herramientas del curso en el ámbito computacional.

3 Competencias a Desarrollar

Competencias genéricas

1. Piensa de forma crítica y analítica.
2. Resuelve problemas de forma estructurada y efectiva.
3. Desarrolla habilidades de investigación y habilidades de comunicación a través de seminarios y presentaciones ante sus colegas.

Competencias específicas

- 1.1 Entiende y domina los fundamentos matemáticos que formalizan la teoría de la computación y los modelos de máquinas computacionales.
- 1.2 Conoce y profundiza los principales modelos computables, sus limitaciones, así como la relación entre lenguajes y cada tipo de autómatas que lo genera.
- 1.3 Comprende los conceptos teóricos y prácticos relacionados con los lenguajes formales.
- 2.1 Aplica métodos y técnicas comunes para la solución de problemas asociados con autómatas finitos, lenguajes formales, funciones computables, computabilidad y recursividad.
- 2.2 Aplica de forma efectiva soluciones computacionales para resolver problemas aplicados relacionados con la teoría de la computación.
- 2.3 Utiliza un enfoque global para resolver problemas. Utiliza herramientas auxiliares en su solución, como lógica, álgebra, expresiones regulares, entre otros.
- 3.1 Desarrolla todas las etapas de una investigación o proyecto aplicado donde se utilizan elementos del autómatas finitos y computabilidad: anteproyecto, formulación de hipótesis, diseño experimental, metodología, resultados y conclusiones.
- 3.2 Escribe un reporte técnico sobre la solución de un problema de interés, teórico o aplicado. Concreta un análisis riguroso y conclusiones importantes.
- 3.3 Comunica de manera efectiva, en forma escrita, oral y visual, los resultados de su investigación.

4 Metodología Enseñanza Aprendizaje

El curso se desarrollará durante diecinueve semanas, con cinco períodos semanales de cuarenta y cinco minutos para desenvolvimiento de la teoría, la resolución de ejemplos y problemas, comunicación didáctica y discusión. Se promoverá el trabajo colaborativo de los estudiantes por medio de listas de ejercicios y proyectos. El curso cuenta con una sesión semanal para la implementación de algoritmos y la práctica de las técnicas del curso.

El resto del curso promoverá la revisión bibliográfica y el auto aprendizaje a través de la solución de los ejercicios del texto, y problemas adicionales, y el desarrollo de un proyecto. Se espera que el alumno desarrolle su trabajo en grupo o individualmente, y que participe activamente y en forma colaborativa durante todo el curso.

5 Contenido

1. Autómatas finitos: Máquinas de estados finitos. Representaciones en formato de grafo y formato tabular. Reglas y protocolos. Ejemplos. Autómatas finitos deterministas (FDA). Notaciones y ejemplos. La función de transición. El lenguaje asociado a un FDA. Autómatas finitos no deterministas (FNA). Notaciones y ejemplos. La función de transición extendida. Equivalencia entre los FDA y los FNA. Aplicaciones: búsqueda de cadenas. Autómatas finitos con transiciones- ε (FNA- ε). Notaciones. Uno de la función de transición. Clausura respecto de ε . Transiciones y lenguajes extendidos para los FNA- ε . Eliminación de las transiciones- ε .
2. Lenguajes y expresiones regulares: *RegExp*. Operadores. y símbolos atómicos. Caracteres ancla, agrupaciones. *Backtracking*. Ejemplos. Alfabetos, cadenas de caracteres y lenguajes formales. Construcción de expresiones regulares. Conversión de los FDA a expresiones regulares. Conversión de expresiones regulares en autómatas. Álgebra y propiedades de las *RegExp*. Aplicaciones: análisis léxico. Propiedades de los lenguajes regulares: *Pumping Lemma*. Clausura respecto de operaciones booleanas. Conversión entre representaciones. Comprobar la pertenencia a un lenguaje regular. Equivalencia y minimización de autómatas: entre estados, entre lenguajes. Minimización de un FDA.
3. Autómatas de pila y lenguajes independientes del contexto (CIL): Autómatas de pila. Lenguajes para un autómata de pila. Analizadores sintácticos $LL(k)$ y $LR(k)$. Equivalencia entre autómatas de autómata a gramática y de gramática a autómata. Complejidad de la conversión. Autómatas de pila deterministas y gramáticas ambiguas. Propiedades de los lenguajes independientes del contexto. La forma normal de Chomsky. Aplicaciones.
4. Máquinas de Turing: Descripción y diagramas de transición. Lenguajes para una máquina de Turing. Técnicas de programación para una máquina de Turing. Extensiones: máquinas con varias cintas, con cintas semi-infinitas, con varias pilas, máquinas contadoras. Simulación. Máquinas de Turing universales. Problemas indecibles para máquinas de Turing. No computabilidad. Teorema de Rice. Semi-computabilidad. Reducciones de Karp, reducciones de Cook. Teorema de Cook. Tesis de Church-Turing.
5. Recursividad: Recursividad y recursividad primitiva. Lenguajes no recursivamente enumerables. Lenguajes recursivos. Funciones recursivas primitivas y μ -recursivas.
6. Complejidad e indecidibilidad: Problemas sobre programas. Problema de correspondencia de Post (PCP). Indecidibilidad del PCP. Indecidibilidad de la ambigüedad de las CIG. Problemas intratables. Las clases P y NP. Ejemplos. El problema de la satisfacibilidad (SAT). Otros problemas NP-completos. Técnicas de reducción de problemas.

6 Bibliografía

Textos:

- J. Hopcroft, R. Motwani, J. Ullman (2007). *Automata Theory, Languages and Computation*. 3rd Edition. Pearson.
- H. Lewis, C. Papadimitriou (1998). *Elements of the Theory of Computation*. 2nd Edition. Prentice-Hall.

Referencias adicionales

- J. G. Brookshear (1993). *Teoría de la Computación, Lenguajes Formales, Autómatas y Complejidad*.
- R. de Castro Corgi (2004). *Teoría de la Computación, Lenguajes Autómatas, Gramáticas*. Unibiblos.
- R. Carrasco Jiménez (2012). *Teoría de lenguajes, gramáticas y autómatas para informáticos*. Unive.
- E. Gaudioso Vásquez (2017). *Introducción a la teoría de autómatas, gramáticas y lenguajes*. URA.
- H. Pedrycz (2022). *Automata Theory and Formal Languages*.

7 Actividades de evaluación

Actividad	Cantidad aproximada	Porcentaje
Listas de ejercicios	8	35%
Pruebas cortas	4	20%
Proyectos	2 ó 3	45%

8 Cronograma

Semana	Tópico	Fecha	Actividades
1	Introducción y motivación al curso. Expresiones regulares.	04-08 julio	Lista 1
2	Autómatas finitos.	11-15 julio	
3		18-22 julio	Lista 2
4		25-29 julio	
5		01-05 agosto	Lista 3
6	Autómatas de pila.	08-12 agosto	
7		15-19 agosto	Lista 4
8		22-26 agosto	
9		29 agosto-02 sept	Lista 5
10	Máquinas de Turing.	05-09 septiembre	
	<i>Semana de asueto</i>	12-16 septiembre	
11		19-23 septiembre	Lista 6
12		26-30 septiembre	
13		03-07 octubre	Lista 7
	Recursividad.		
14		10-14 octubre	
15		17-21 octubre	Lista 8
16	Entrega de proyectos.	24-28 octubre	Proyecto
17	Decidibilidad.	31 octubre-04 nov	
18		07-11 noviembre	
19		14-18 noviembre	
20	Presentación de proyectos.	21-25 noviembre	Proyecto