



Universidad Nacional de Ingeniería (UNI)

Escuela Profesional de
Ciencia de la Computación
Sílabo 2024-II

1. CURSO

CS211. Theory of Computation (Mandatory)

2. INFORMACIÓN GENERAL

2.1 Curso	: CS211. Theory of Computation
2.2 Semestre	: 4 th Semester.
2.3 Créditos	: 4
2.4 horas	: 2 HT; 4 HP;
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Mandatory
2.7 Modalidad de aprendizaje	: Face to face
2.8 Prerrequisitos	: CS1D2. Discrete Structures II. (2 nd Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

This course emphasizes formal languages, computer models and computability, as well as the fundamentals of computational complexity and complete NP problems.

5. OBJETIVOS

- That the student learn the fundamental concepts of the theory of formal languages.

6. RESULTADOS DEL ESTUDIANTE

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

7. TEMAS

Unidad 1: Computabilidad y complejidad básica de autómatas (20 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Máquinas de estado finito. • Expresiones regulares. • Problema de la parada. • Gramáticas libres de contexto. • Introducción a las clases P y NP y al problema P vs. NP. • Introducción y ejemplos de problemas NP- Completos y a clases NP-Completos. • Máquinas de Turing, o un modelo formal equivalente de computación universal. • Máquinas de Turing no determinísticas. • Jerarquía de Chomsky. • La tesis de Church-Turing. • Computabilidad. • Teorema de Rice. • Ejemplos de funciones no computables. • Implicaciones de la no-computabilidad.	• Discute el concepto de máquina de estado finito [Evaluar] • Diseñe una máquina de estado finito determinista para aceptar un determinado lenguaje [Evaluar] • Genere una expresión regular para representar un lenguaje específico [Evaluar] • Explique porque el problema de la parada no tiene solución algorítmica [Evaluar] • Diseñe una gramática libre de contexto para representar un lenguaje especificado [Evaluar] • Defina las clases P y NP [Evaluar] • Explique el significado de NP-Complejidad [Evaluar] • Explique la tesis de Church-Turing y su importancia [Familiarizarse] • Explique el teorema de Rice y su importancia [Familiarizarse] • Da ejemplos de funciones no computables [Familiarizarse] • Demuestra que un problema es no computable al reducir un problema clásico no computable en base a él [Familiarizarse]
Lecturas : [Mar10], [Lin11], [Sip12]	

Unidad 2: Complejidad Computacional Avanzada (20 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Revisión de las clases P y NP; introducir espacio P y EXP. • Jerarquía polinomial. • NP completitud (Teorema de Cook). • Problemas NP completos clásicos. • Técnicas de reducción.	• Defina la clase P-Space y su relación con la clase EXP [Evaluar] • Defina la clase P-Space y su relación con la clase EXP [Evaluar] • Explique el significado de NP-Completo (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar] • Muestre ejemplos de problemas clásicos en NP - Completo [Evaluar] • Pruebe que un problema es NP- Completo reduciendo un problema conocido como NP-Completo [Evaluar]
Lecturas : [Mar10], [Lin11], [Sip12], [HU13]	

Unidad 3: Teoría y Computabilidad Avanzada de Autómatas (20 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Conjuntos y Lenguajes: – Lenguajes Regulares. – Revisión de autómatas finitos determinísticos (Deterministic Finite Automata DFAs) – Autómata finito no determinístico (Nondeterministic Finite Automata NFAs) – Equivalencia de DFAs y NFAs. – Revisión de expresiones regulares; su equivalencia con autómatas finitos. – Propiedades de cierre. – Probando no-regularidad de lenguajes, a través del lema de bombeo (Pumping Lemma) o medios alternativos. • Lenguajes libres de contexto: – Autómatas de pila (Push-down automata PDAs) – Relación entre PDA y gramáticas libres de contexto. – Propiedades de los lenguajes libres de contexto.	• Determina la ubicación de un lenguaje en la jerarquía de Chomsky (regular, libre de contexto, enumerable recursivamente) [Evaluar] • Convierte entre notaciones igualmente poderosas para un lenguaje, incluyendo entre estas AFDs, AFNDs, expresiones regulares, y entre AP y GLCs [Evaluar]
Lecturas : [HU13], [Bro93]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.
- [Mar10] John Martin. *Introduction to Languages and the Theory of Computation*. 4th. McGraw-Hill, 2010.
- [Lin11] Peter Linz. *An Introduction to Formal Languages and Automata*. 5th. Jones & Bartlett Learning, 2011.
- [Sip12] Michael Sipser. *Introduction to the Theory of Computation*. 3rd. Cengage Learning, 2012.
- [HU13] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Pearson Education, 2013.