

PROGRAMA DE CURSO

Código	Nombre			
MA5201	Calculabilidad y Complejidad Computacional			
Nombre en Inglés				
Calculability and Computational Complexity				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1.5	5.5
Requisitos			Carácter del Curso	
Optimización combinatorial MA4701			Obligatorio	
Resultados de Aprendizaje				
<p>Conocer los elementos básicos de la teoría de la computación: autómatas finitos, algoritmos, reducciones, recursos.</p> <p>Conocer las técnicas para la clasificación de problemas computacionales de acuerdo al uso de recursos.</p> <p>Conocer clases de complejidad de tiempo, de espacio, aleatorias y definidas a través de protocolos interactivos.</p> <p>Dar demostraciones formales de afirmaciones propias a la teoría desarrollada.</p>				

En el formulario se solicita el número de créditos académicos SCT - "Sistema de Créditos Transferibles de Chile". Este sistema fue adoptado por la Universidad de Chile y por el resto de las universidades miembros del Consejo de Rectores.

Un crédito SCT equivale a la proporción respecto de la carga total de trabajo necesaria para completar un año de estudios a tiempo completo. Se ha convenido que el trabajo anual tienda a los 60 créditos; en el caso de los programas de estudio de nuestra Facultad 1 U.D. equivale a 0,6 créditos SCT. Por ejemplo, un curso de 10 U.D. equivale a 6 créditos SCT

Metodología Docente	Evaluación General
Clases presenciales, dictadas por el profesor y clases auxiliares de resolución de problemas.	3 controles y un examen.

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Autómatas y lenguajes formales.	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<p>Autómatas finitos, lenguajes regulares, no determinismo, lema del bombeo.</p> <p>Autómatas de pila, lenguajes libres de contexto, lema del bombeo.</p>	<p>Conocer y aplicar las herramientas de la unidad.</p> <p>Demostrar afirmaciones acerca de los temas de la unidad.</p>	3,4,5

Número	Nombre de la Unidad	Duración en Semanas
2	Calculabilidad.	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<p>La máquina de Turing y la tesis de Turing-Church.</p> <p>Decidibilidad, indecidibilidad, reducciones.</p> <p>El teorema de Rice y el teorema de recursión.</p> <p>Indecidibilidad y lógica: el teorema de incompletitud de Godel.</p>	<p>Conocer y aplicar las herramientas de la unidad.</p> <p>Demostrar afirmaciones acerca de los temas de la unidad.</p>	3,4,5

Número	Nombre de la Unidad	Duración en Semanas
3	Complejidad computacional.	5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<p>Complejidad de tiempo: las clases P, NP, EXP, NEXP.</p> <p>Reducciones, completitud y dificultad; problemas clásicos.</p> <p>P versus NP: la clase co-NP, relativización, alternancia, la jerarquía polinomial.</p> <p>Complejidad de espacio: las clases L, NL, PSPACE.</p> <p>Nodeterminismo y Teorema de Savitch. Igualdad $L=NL$.</p> <p>Funciones constructibles y jerarquías de tiempo y de espacio.</p>	<p>Conocer y aplicar las herramientas de la unidad.</p> <p>Demostrar afirmaciones acerca de los temas de la unidad.</p>	1,2,3,4,5

Número	Nombre de la Unidad	Duración en Semanas
4	Temas avanzados	4
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<p>Clases de Complejidad probabilistas: BPP, RP, ZPP, PP, AM, MA.</p> <p>Demostraciones interactivas y su relación con el uso del espacio: $PSPACE=IP$.</p> <p>Demostraciones interactivas y aproximabilidad.</p> <p>Implicancias algorítmicas del Teorema de los menores.</p>	<p>Conocer y aplicar las herramientas de la unidad.</p> <p>Demostrar afirmaciones acerca de los temas de la unidad.</p>	1,4,5

Bibliografía

1. S. Arora y B. Barak, Computational Complexity, 2009.
2. M. Garey y D. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W. H. Freeman and Company, 1979.
3. J. Hopcroft y J. Ullman, Introduction to automata theory, languages and computation, Addison Wesley, 1979.
4. C. Papadimitriou, Computational Complexity, Addison Wesley, 1994.
5. M. Sipser, Introduction to the theory of computation, PWS Publishing Company, 1997.

OTRAS REFERENCIAS.

1. J. Balcázar, J. Díaz y J. Gabarró, Structural Complexity I, Springer, segunda edición, 1995.
2. D. Bovet y P. Crescenzi. Introduction to the Theory of Complexity. Prentice Hall, 1993.
3. B. Moret, Theory of Computation, Addison-Wesley, 1998.
4. N. Pippenger, Theories of Computability, Cambridge Univ Press, 1997.
5. J. Savage. Models of Computation: Exploring the power of computing. Addison-Wesley, 1998.
6. J. van Leeuwen (ed.), Handbook of Theoretical Computer Science, Elsevier Science, Amsterdam, 1990.

Vigencia desde:	OTOÑO 2011
Elaborado por:	GRUPO MATEMATICAS DISCRETAS (M. Matamala)