FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

PROGRAMA DE CURSO

Código	Nombre			
CC3102	Teoría de la Computación			
Nombre en Inglés				
Theory of Computation				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1.5	5.5
Requisitos			Carácter del Curso	
CC3001, CC3101			Obligatorio	

Resultados de Aprendizaje

El curso entrega los conceptos fundamentales de la Computabilidad y la Complejidad Computacional, así como de Lenguajes Formales, considerando en particular lenguajes regulares y libres del contexto.

El alumno que apruebe el curso habrá demostrado:

- Comprender lo que es un lenguaje regular y dominar sus representaciones en forma de autómatas finitos y expresiones regulares. Conocer las formas más elementales de demostrar que un lenguaje no es regular. Comprender las aplicaciones prácticas de los autómatas a búsqueda en texto.

- Comprender lo que es un lenguaje libre del contexto y dominar sus representaciones en forma de autómatas de pila y gramáticas libres del contexto. Conocer las formas más elementales de demostrar que un lenguaje no es libre del contexto. Comprender las aplicaciones prácticas de esta teoría al parsing.

- Comprender que existen problemas que no se pueden resolver por computador, la diferencia entre lenguaje decidibles y aceptables, y conocer algunos ejemplos emblemáticos, en particular el problema de la detención.

- Comprender el formalismo de Máquinas de Turing y utilizarlo como modelo de computación. Comprender la Tesis de Church y tener noción de algunos formalismos alternativos.

- Comprender el concepto de complejidad computacional y el significado de las clases de problemas P y NP. Comprender el concepto de NP-completitud y conocer varios de los problemas NP-completos más famosos. Comprender el mecanismo para establecer que un problema es NP-completo.

Metodología Docente	Evaluación General
Clases expositivas del profesor de cátedra, buscando la participación de los alumnos en pequeños problemas que se van proponiendo durante la exposición. Clases auxiliares dedicadas a explicar ejemplos más extensos, resolver ejercicios propuestos, y preparación pre y post controles.	Tres controles evaluando ejercicios simples y un examen final evaluando lo aprendido con más perspectiva. NC = promedio ponderado del examen (40%) y del promedio de los controles (60%). Tareas tanto teóricas (resolución de ejercicios) como prácticas, que permiten al alumno comprender la conexión con las aplicaciones prácticas.

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Lenguajes Regulares	3

Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Alfabetos, cadenas y lenguajes. Representación finita de lenguajes. 2. Autómatas finitos determinísticos. 3. Autómatas finitos no determinísticos. 4. Equivalencia entre ambos tipos de autómatas. 5. Expresiones regulares. 6. Equivalencia entre expresiones regulares y autómatas. 7. Propiedades de clausura y algorítmicas. 8. Lema del bombeo.	Comprender lo que es un lenguaje regular y dominar sus representaciones en forma de autómatas finitos y expresiones regulares. Conocer las formas más elementales de demostrar que un lenguaje no es regular. Comprender las aplicaciones prácticas de los autómatas a búsqueda en texto.	[1] Cap. 2 [2] Cap. 1 y 2 (2.1 – 2.4) [4] Cap. 1

Número	Nombre de la Unidad	Duración en Semanas
2	Lenguajes Libres del Contexto	3

Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Gramáticas libres del contexto. 2. Autómatas de pila. 3. Equivalencia entre gramáticas y autómatas. 4. Propiedades de clausura y algorítmicas. 5. Teorema de bombeo. 6. Determinismo y parsing.	Comprender lo que es un lenguaje libre del contexto y dominar sus representaciones en forma de autómatas de la pila y gramáticas libres del contexto. Conocer las formas más elementales de demostrar que un lenguaje no es libre del contexto. Comprender las aplicaciones prácticas de esta teoría al parsing.	[1] Cap. 3 [2] Cap. 3 (3.7 sólo parte) [4] Cap. 2

Número	Nombre de la Unidad	Duración en Semanas
3	Máquinas de Turing	2.5

Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Definición de Máquinas de Turing (MTs). Configuraciones y modelo de computación. 2. Modularización y solución de problemas más complejos usando MTs. Uso de MTs para decidir lenguajes, aceptar lenguajes, y calcular funciones. 3. Extensiones de MTs: múltiples cintas y otras. Simulaciones. 4. MTs no determinísticas y su simulación.	Comprender el formalismo de Máquina de Turing y utilizarlo como un modelo de computación. Comprender las nociones de lenguajes decidibles y aceptables.	[1] Cap. 4 [2] Cap. 4 (4.1 – 4.3 y 4.5) [4] Cap. 3

Número	Nombre de la Unidad	Duración en Semanas
4	Computabilidad	3.5

Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. La Máquina Universal de Turing. Simulación. 2. La Tesis de Church y otros formalismos. Equivalencia con el modelo RAM. 3. El problema de la detención. Lenguajes decidibles y aceptables. 4. Otros problemas indecidibles acerca de MTs. Reducción de problemas. 5. Gramáticas dependientes del contexto. Equivalencia con MTs. Enumerabilidad. 6. Problemas indecidibles acerca de gramáticas. Otros problemas indecidibles.	Comprender que existen problemas que no se pueden resolver por computador, la diferencia entre lenguajes decidibles y aceptables, y conocer algunos ejemplos emblemáticos, en particular el problema de la detención. Comprender la Tesis de Church y tener noción de algunos formalismos alternativos.	[1] Cap. 5 [2] Cap. 4.6 y 5 [4] Cap. 4 y 5

Número	Nombre de la Unidad	Duración en Semanas
5	Complejidad Computacional	3

Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Longitud de una computación. Lenguajes y problemas. Complejidad de un problema. Notación O. 2. Las clases P y NP. Reducción polinomial. NP-completitud. 3. El problema NP-completo de satisfactibilidad de fórmulas booleanas. 4. Otros problemas NP-completos: clique, recubrimiento de vértices circuito hamiltoniano, coloreo y otros.	Comprender el concepto de complejidad computacional y el significado de las clases de problema P y NP. Comprender el concepto de NP-completitud y conocer varios de los problemas NP-completos más famosos. Comprender el mecanismo para establecer que un problema es NP-completo.	[1] Cap. 6 [2] Cap. 6 y 7.1 [3] Cap. 34 [4] Cap. 7

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

Bibliografía

[1] G. Navarro. Teoría de la Computación. Apuntes y Ejercicios, 2006 (actualización 2016) www.dcc.uchile.cl/gnavarro/apunte.html

[2] H. Lewis, C. Papadimitriou. Elements of the Theory of Computation, 2nd edition. Prentice-Hall, 1998.

[3] T. Cormen. C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms, 3rd edition. MIT Press, 2009.

[4] M. Sipser. Introduction to the Theory of Computation, 3rd edition. Cengage Learning, 2013.

[5] D. Kelley. Teoría de Autómatas y Lenguajes Formales. Prentice-Hall, 1995.

[6] J. Hopcroft, R. Motwani, J. Ullman. Introduction to Automata Theory, Languages and Computation, 3rd edition. Pearson Education, 2014.

[7] A. Aho, J. Hopcroft, J. Ullman. The design and Analysis of Computer Algorithms. Addison-Wesley, 1974.

Vigencia desde:	Última revisión: Primavera 2016
Elaborado por:	Gonzalo Navarro