



Teaching Guide

Identifying Data					2015/16
Subject (*)	Teoría de Autómatas e Linguaxes Formais		Code	614111301	
Study programme	Enxeñeiro en Informática				
Descriptors					
Cycle	Period	Year	Type	Credits	
First and Second Cycle	1st four-month period	Third	Troncal	7.5	
Language	SpanishGalician				
Teaching method	Face-to-face				
Prerequisites					
Department	Computación				
Coordinador	Blanco Ferro, Antonio angel	E-mail	antonio.blanco.ferro@udc.es		
Lecturers	Blanco Ferro, Antonio angel	E-mail	antonio.blanco.ferro@udc.es		
Web	http://www.dc.fi.udc.es/~grana/TALF/				
General description	Se trata de una asignatura troncal, que se imparte de manera cuatrimestral en el tercer curso de la titulación de Ingeniero en Informática. Destaca el carácter integrador de su contenido, ya que sirve de puente entre lo que podemos denominar una "visión de usuario" de los lenguajes informáticos, representada por la programación estándar, y una "visión generativa" de éstos, en la que el alumno construye y adecúa un lenguaje de programación en atención a sus requerimientos. Finalmente, se transmite también al alumno una visión formal de los fundamentos propios de la ciencia de la computación.				

Study programme competences / results

Code	Study programme competences / results
A1	Aprender de maneira autónoma novos coñecementos e técnicas avanzadas axeitadas para a investigación, o deseño e o desenvolvemento de sistemas e servizos informáticos.
A3	Concibir e planificar o desenvolvemento de aplicacións informáticas complexas ou con requisitos especiais.
B1	Aprender a aprender.
B2	Resolver problemas de forma efectiva.
B3	Aplicar un pensamento crítico, lóxico e creativo.
B4	Aprendizaxe autónoma.
B5	Traballar de forma colaborativa.
B8	Traballar en equipos de carácter interdisciplinar.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.

Learning outcomes

Learning outcomes	Study programme competences / results		
Conocer en profundidad la estructura y función de los sistemas de descripción y reconocimiento de lenguajes formales.	A1	B1 B4	
Estudiar los conceptos, modelos y técnicas relacionados con estas cuestiones.	A1	B1 B4	
Conocer las estructuras de datos y los algoritmos utilizados para implementar los distintos modelos de reconocimiento de lenguajes formales, así como sus posibles dominios de aplicación práctica.	A1 A3	B1 B4	C6
Realizar implementaciones de estos modelos en alguno de esos dominios.	A1 A3	B2 B3 B5	C6
Sintetizar todos los conceptos estudiados en ideas concretas que permitan comprender mejor los fundamentos de la computación	A1	B1 B4	



Perfeccionar las habilidades para realizar futuros trabajos de análisis, diseño y programación.	A1 A3	B2 B3 B5	C6
Considerar la integración de las técnicas y estructuras estudiadas aquí en otros dominios de aplicación.	A1 A3	B2 B3 B5 B8	C6

Contents	
Topic	Sub-topic
Preliminares matemáticos	Lógica elemental Teoría de conjuntos Relaciones y funciones Inducción matemática Cardinalidad
Lenguajes formales	Alfabetos, palabras y lenguajes Operaciones con palabras Operaciones con lenguajes
Lenguajes regulares y autómatas finitos	Lenguajes sobre alfabetos Lenguajes regulares y expresiones regulares Autómata finito determinista (AFD) Autómata finito no determinista (AFN) Equivalencia entre AFNs y AFDs Autómata finito con epsilon transiciones Autómatas finitos y expresiones regulares Aplicaciones prácticas de las expresiones regulares y de los autómatas finitos
Lenguajes independientes del contexto y autómatas de pila	Gramáticas regulares Gramáticas regulares y lenguajes regulares Gramáticas independientes del contexto Árboles de derivación y ambigüedad Simplificación de gramáticas independientes del contexto Propiedades de los lenguajes independientes del contexto Algoritmos de análisis sintáctico Autómatas de pila Forma normal de Greibach
Máquinas de Turing	Definiciones básicas Máquinas de Turing como aceptadoras de lenguajes Construcción de máquinas de Turing Modificaciones de las máquinas de Turing Máquina de Turing universal
Lenguajes recursivamente enumerables	Lenguajes aceptados por máquinas de Turing Lenguajes regulares e independientes del contexto como lenguajes recursivos Propiedades de los lenguajes recursivos y recursivamente enumerables Gramáticas no restringidas y lenguajes recursivamente enumerables Lenguajes sensibles al contexto y la jerarquía de Chomsky
Resolubilidad	El problema de la parada El problema de correspondencia de Post Problemas no decidibles en lenguajes independientes del contexto



Computabilidad	Fundamentos de la teoría de funciones recursivas Alcance de las funciones recursivas primitivas Funciones recursivas parciales El poder de los lenguajes de programación
Introducción a la teoría de la complejidad computacional	Complejidad algorítmica Modelo general de cómputo y complejidad computacional Tiempo y espacio en máquinas de Turing Las distintas clases de complejidad Los problemas tratables y no tratables Reducibilidad en tiempo polinómico Problemas NP-completos

Planning				
Methodologies / tests	Competencies / Results	Teaching hours (in-person & virtual)	Student?s personal work hours	Total hours
Guest lecture / keynote speech		30	60	90
Laboratory practice		10	20	30
Multiple-choice questions		4	4	8
Supervised projects		1	5	6
Seminar		3	0	3
Collaborative learning		4	4	8
Problem solving		3	16	19
Long answer / essay questions		3	16	19
Personalized attention		4.5	0	4.5

(*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.

Methodologies	
Methodologies	Description
Guest lecture / keynote speech	La técnica que mejor se adapta a la impartición de los contenidos teóricos de esta asignatura está constituida por las clases magistrales. En ellas, haremos un uso intensivo de la pizarra y de las transparencias, de modo que el ritmo de exposición de conceptos por parte del profesor y el de asimilación de los mismos por parte del alumno sean lo más acordes posible.
Laboratory practice	Las prácticas de laboratorio tendrán horas de laboratorio reservadas, con ordenadores a disposición de los alumnos. Estas horas serán utilizadas para implementar en algún lenguaje de programación los algoritmos más destacados, de entre todos aquéllos que hayan sido presentados en las sesiones teóricas.
Multiple-choice questions	Se realizarán controles tipo test al final de cada bloque temático, que permitirán al profesor conocer el grado de asimilación de la materia por parte de los alumnos, y modificar la estrategia docente si es necesario.
Supervised projects	Los trabajos de grupos autónomos tutelados se realizarán a lo largo de todo el cuatrimestre. El profesor elegirá un tema de trabajo que será asignado por igual a todos los grupos. El tema será presentado por el profesor en una sesión en el aula, será desarrollado por los alumnos en horas no presenciales, y será supervisado y evaluado por el profesor en las tutorías en grupo. La evaluación se realizará a partir de la exposición de una memoria final por parte de los alumnos.
Seminar	Los seminarios se implementarán bajo la forma de un ciclo de charlas cortas o conferencias, sobre aplicaciones prácticas relacionadas con la materia de la asignatura. El objetivo de estas charlas es el de completar la percepción general que el alumno tiene sobre cómo los conceptos vistos en clase son puestos en práctica en la vida real.



Collaborative learning	Los trabajos de grupos cooperativos se realizarán utilizando la "técnica puzzle" de Aronson. Según esta técnica, el profesor elige previamente un tema de trabajo y divide a los alumnos en varios grupos. A cada miembro de cada grupo se le asigna una sección de dicho tema, para que la estudie y la comente con los miembros de los otros grupos a los que les ha sido asignada la misma sección. Posteriormente, cada alumno regresa a su grupo y explica al resto de miembros su sección. Finalmente, el profesor realiza a los alumnos un test general sobre el tema elegido. Dicho test permitirá al profesor no sólo evaluar el grado de comprensión de los nuevos conocimientos adquiridos, sino también el nivel de cooperación que ha tenido lugar entre los miembros de cada grupo concreto.
Problem solving	Se pondrán a disposición de los alumnos una serie de boletines de ejercicios, correspondientes a los bloques temáticos del programa de la asignatura. Los alumnos deberán entregar al profesor sus soluciones personales a estos ejercicios. El profesor deberá corregirlas, evaluarlas y comentarlas durante al menos una sesión en el aula.
Long answer / essay questions	Se implementará bajo la forma de un examen final escrito.

Personalized attention

Methodologies	Description
Laboratory practice Supervised projects	<p>Dado el carácter personalizado de las prácticas de laboratorio, de los trabajos tutelados y de las tutorías, estas actividades no deben dedicarse a extender los contenidos con nuevos conceptos, sino a aclarar los conceptos ya expuestos.</p> <p>El profesor debe además utilizarlas como una interacción que le permita extraer conclusiones respecto al grado de asimilación de la materia por parte de los alumnos.</p> <p>De esta manera, podrá desarrollar las clases magistrales y el resto de actividades no personalizadas atendiendo al progreso de los alumnos en las capacidades de comprensión y asimilación de los contenidos impartidos, compaginando el avance general de la clase con una atención específica a aquellos alumnos que presenten mayores dificultades en la tarea del aprendizaje y con un apoyo adicional a aquellos otros que presenten mayor desenvoltura y deseen ampliar conocimientos.</p>

Assessment

Methodologies	Competencies / Results	Description	Qualification
Laboratory practice		Implementación de algoritmos en algún lenguaje de programación	25
Multiple-choice questions		Controles tipo test	7.5
Supervised projects		Trabajo de grupos autónomos tutelados	5
Collaborative learning		Trabajo de grupos cooperativos	7.5
Problem solving		Boletines de ejercicios	5
Long answer / essay questions		Examen final escrito	50
Others			

Assessment comments

En el examen final se requiere una nota mínima de 3 puntos (sobre 10).
--

Sources of information

--



Basic	<ul style="list-style-type: none">- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2002). Introducción a la teoría de autómatas, lenguajes y computación. Addison Wesley- Thomas A. Sudkamp (1988). Languages and machines: an introduction to the theory of computer science. Addison Wesley- Dean Kelley (1995). Teoría de autómatas y lenguajes formales. Prentice Hall
Complementary	<ul style="list-style-type: none">- Harry R. Lewis, Christos H. Papadimitriou (1998). Elements of the theory of computation. Prentice Hall- Peter J. Denning, Jack B. Dennis, Joseph E. Qualitz (1978). Machines, languages and computation. Prentice Hall- J. Glenn Brookshear (1993). Teoría de la computación: lenguajes formales, autómatas y complejidad. Addison Wesley Iberoamericana

Recommendations

Subjects that it is recommended to have taken before

Intelixencia Artificial/614111404
Compiladores/614111405
Linguaxes Naturais/614111625

Subjects that are recommended to be taken simultaneously

Subjects that continue the syllabus

Estrutura de Datos e da Información/614111102
Álxebra/614111106
Matemática Discreta I/614111107
Programación/614111109
Algoritmos/614111206
Programación Declarativa/614111207

Other comments

(*)The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.