

Presentación

Teoría de la Computación

Grado en Ingeniería Informática

Contenidos

- 1 Descripción de la asignatura
- 2 Temario
- 3 Planificación y evaluación
- 4 Bibliografía

Contenidos

- 1 Descripción de la asignatura
- 2 Temario
- 3 Planificación y evaluación
- 4 Bibliografía

Descripción de la asignatura

Tradicionalmente, se considera que la vertiente teórica de las ciencias de la computación ha evolucionado a partir de tres disciplinas distintas: la **matemática**, la **ingeniería** y la **lingüística**.

- Las **raíces matemáticas** datan de principios del siglo XX (con los problemas de **Hilbert** y los teoremas de incompletud de **Gödel**) y culminan en los años treinta con la modelización del concepto de algoritmo y con el establecimiento por parte de **Turing** de los límites fundamentales de la computación.

Para ello, Turing se basó en su conocido **modelo de máquina abstracta**, muy similar a la idea que hoy tenemos de un computador, aunque los verdaderos computadores aún tardarían varios años en llegar (el propio Turing estuvo involucrado en la construcción de computadores electromecánicos durante la II Guerra Mundial).

Descripción de la asignatura

El éxito del planteamiento de Turing sigue vigente hoy en día, ya que fue corroborado por otros trabajos posteriores orientados hacia la formulación de modelos de computación alternativos: las funciones recursivas de Kleene, el λ -cálculo de Church y los sistemas de correspondencia de Post.

Todos ellos se han mostrado formalmente equivalentes al modelo de Turing, es decir, computan exactamente las mismas funciones y reconocen los mismos lenguajes.

La conjetura no demostrada de que cualquier forma general de computación nos permitirá calcular únicamente las funciones recursivas parciales (o, lo que es lo mismo, lo que las máquinas de Turing o los ordenadores modernos pueden calcular) se conoce como *Hipótesis de Church* o *Tesis de Church-Turing*.

- Los **orígenes en ingeniería** pueden localizarse también en la segunda mitad de los años treinta, con la observación por parte de **Shannon** de que las funciones de los circuitos combinatorios y secuenciales (construidos al principio con relés y luego con dispositivos electrónicos de vacío y de estado sólido) podían representarse mediante la notación simbólica de las álgebras de **Boole**.

Más tarde, en los años cincuenta, **Huffman** y **Caldwell** continuaron este trabajo inicial para obtener una formalización de los circuitos secuenciales.

Estos trabajos evolucionaron hasta derivar en lo que hoy conocemos como la *teoría de máquinas de estado finito*.

- La **contribución lingüística** se debe en origen a **Chomsky**.

Sus trabajos de finales de los años cincuenta representan los primeros pasos en el **estudio de los lenguajes formales**, y deben entenderse en el marco del tratamiento informático de los lenguajes naturales.

En concreto, Chomsky redujo el estudio de los lenguajes al estudio de la estructura de las frases y del significado de las mismas, clasificando los lenguajes en cuatro clases según la estructura de sus reglas gramaticales.

Es lo que hoy en día conocemos como la ***Jerarquía de Chomsky***.

En resumen:

- Se trata de una asignatura en la que 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**.

Contenidos

- 1 Descripción de la asignatura
- 2 Temario**
- 3 Planificación y evaluación
- 4 Bibliografía

Introducción a la teoría de lenguajes formales

- Tema 0: Preliminares matemáticos
- Tema 1: Lenguajes formales

Formalismos para la descripción y el reconocimiento de lenguajes

- Tema 2: Lenguajes regulares y autómatas finitos
- Tema 3: Lenguajes independientes del contexto y autómatas de pila

Máquinas de Turing y lenguajes

- Tema 4: Máquinas de Turing
- Tema 5: Lenguajes recursivamente enumerables

Nociones generales sobre resolubilidad y computabilidad

- Tema 6: Resolubilidad
- Tema 7: Computabilidad

Bloque I:

- Tema 0: Preliminares matemáticos
- Tema 1: Lenguajes formales
- Tema 2: Lenguajes regulares y autómatas finitos

Bloque II:

- Tema 3: Lenguajes independientes del contexto y autómatas de pila

Bloque III:

- Tema 4: Máquinas de Turing
- Tema 5: Lenguajes recursivamente enumerables
- Tema 6: Resolubilidad

Tema 0: Preliminares matemáticos

- Lógica elemental
- Definiciones básicas sobre teoría de conjuntos
- Operaciones con conjuntos
- Relaciones y funciones
- Inducción matemática
- Cardinalidad

Tema 1: Lenguajes formales

- Alfabetos, palabras y lenguajes
- Operaciones con palabras
- Operaciones con lenguajes

Tema 2: Lenguajes regulares y autómatas finitos

- Lenguajes sobre alfabetos
- Lenguajes regulares y expresiones regulares
- Autómata finito determinista (AFD)
- Autómata finito no determinista (AFN)
- Equivalencia entre AFN,s y AFD,s
- Autómata finito no determinista con ϵ -transiciones (AFN- ϵ)
- Autómatas finitos y expresiones regulares
- Propiedades de los lenguajes regulares
- Aplicaciones prácticas de las expresiones regulares y de los autómatas finitos

Tema 3: Lenguajes independientes del contexto y autómatas de pila

- 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

Tema 4: 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

Tema 5: 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

Tema 6: Resolubilidad

- El problema de la parada
- El problema de correspondencia de Post
- Problemas no decidibles en lenguajes independientes del contexto

Tema 7: Computabilidad

- Fundamentos de la teoría de funciones recursivas
- Alcance de las funciones recursivas primitivas
- Funciones recursivas parciales
- El poder de los lenguajes de programación

Contenidos

- 1 Descripción de la asignatura
- 2 Temario
- 3 Planificación y evaluación**
- 4 Bibliografía

Planificación:

- Docencia expositiva:
 - Clases magistrales: 12 sesiones de $1\frac{1}{2}$ horas
 - Controles eliminatorios: 3 sesiones de $1\frac{1}{2}$ horas.
- Docencia interactiva:
 - Prácticas de laboratorio: 10 sesiones de $1\frac{1}{2}$ horas.
- Trabajos en grupos reducidos:
 - Resolución de ejercicios: 5 sesiones de $1\frac{1}{2}$ horas.

Evaluación:

- Controles eliminatorios / Examen final: 6 puntos.
- Prácticas de laboratorio: 3 puntos.
- Boletines de ejercicios: 1 punto.

Contenidos

- 1 Descripción de la asignatura
- 2 Temario
- 3 Planificación y evaluación
- 4 Bibliografía**

- Dean Kelley.
Teoría de autómatas y lenguajes formales.
Prentice Hall, 1995.
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman.
Introducción a la teoría de autómatas, lenguajes y computación.
Addison Wesley, 2002.
- Thomas A. Sudkamp
Languages and machines: an introduction to the theory of computer science.
Addison Wesley, 1988.