11-12

GUÍA DE ESTUDIO DE LDI



PROGRAMACION DECLARATIVA

CÓDIGO 0153309-

UNED

11-12

PROGRAMACION DECLARATIVA

CÓDIGO 0153309-

ÍNDICE

OBJETIVOS
CONTENIDOS
EQUIPO DOCENTE
BIBLIOGRAFÍA BÁSICA
BIBLIOGRAFÍA COMPLEMENTARIA
SISTEMA DE EVALUACIÓN
HORARIO DE ATENCIÓN AL ESTUDIANTE

OBJETIVOS

La asignatura tiene como objetivo dar a los alumnos una visión general del paradigma declarativo (programación funcional y programación lógica) y de las características más relevantes de los lenguajes actuales (que, para muchos usos pueden sustituir con ventaja a los lenguajes imperativos convencionales). Se enseñan también las técnicas de programación básicas de estos lenguajes.

Los lenguajes imperativos se basan, al igual que el código máquina, en el modelo Von Neumann, en el que un programa detalla la secuencia exacta de instrucciones que el computador ha de seguir para modificar el estado de la memoria. Las instrucciones típicas son las que modifican el estado del cómputo (como las asignaciones) y las que gobiernan el flujo de control (iteraciones, condicionales).

Los lenguajes declarativos, por el contrario, están más cercanos a la forma de pensar del programador (son de más *alto nivel*), y no necesitan la especificación de la secuencia en la que tienen lugar las operaciones elementales dentro de la máquina. El tamaño de los programas y el tiempo necesario para desarrollarlos son sensiblemente menores, en general, que los correspondientes en cualquier lenguaje imperativo. Como contrapartida, los programas desarrollados en lenguajes funcionales suelen ser más ineficientes que los programas imperativos y lenguajes máquina: lo que se gana en transparencia y en tiempo de desarrollo, se pierde en eficiencia. Hasta hace poco, la diferencia estribaba en que existían compiladores muy buenos para lenguajes imperativos que permiten su uso a escala industrial, pero no para lenguajes declarativos, considerados como productos de “laboratorio”. En la actualidad están empezando a usarse lenguajes lógicos y funcionales para aplicaciones y proyectos a gran escala, y es posible que en un futuro no muy lejano lenguajes como los funcionales sean preferidos por una mayoría de programadores.

RELACIÓN CON OTRAS ASIGNATURAS

La asignatura de Programación orientada a la Inteligencia Artificial (POIA) se solapa parcialmente con esta asignatura, aunque con una visión diferente. POIA se focaliza sobre problemas que requieren típicamente técnicas de Inteligencia Artificial, mientras que aquí aplicaremos este paradigma a problemas informáticos en diversos dominios y tipos de tareas. Por ejemplo, tiene relación estrecha tanto con métodos formales en ingeniería del software como con las bases de datos. Por otra parte son complementarias, ya que la programación funcional abarca distintas familias; en esta asignatura se aborda el estudio de lenguajes con *disciplina de tipos* mientras que POIA se centra en el Lisp, lenguaje que carece de la misma.

CONTENIDOS

Unidad Didáctica I. Programación funcional

La programación funcional es una forma de programar que pone el énfasis en la evaluación de expresiones en lugar de en la ejecución de órdenes. La ejecución de un programa funcional no es sino la aplicación de una función a unos datos; la expresión que resulta se reduce, mediante re-escrituras sucesivas, a expresiones equivalentes más sencillas, hasta llegar a una expresión irreducible que se considera el resultado de la ejecución del

programa. Una característica esencial de los lenguajes funcionales es que pueden existir funciones de orden superior, es decir, funciones que tienen como argumento a otras funciones (dicho de otra forma, programas que tienen como argumentos otros programas). Otra característica esencial es la *evaluación perezosa* (sólo se realiza un cálculo cuando otro cálculo posterior lo necesita), que permite la definición de estructuras infinitas de datos, así como de funciones y tipos de datos no estrictos.

Los lenguajes funcionales han sido, desde la invención del *Lisp*, los más utilizados en Inteligencia Artificial. En los últimos años, la consolidación de estándares como el Haskell y la aparición de compiladores cada vez más eficientes han hecho que se estén comenzando a utilizar en aplicaciones industriales.

Unidad Didáctica II. Programación lógica

Por programación lógica se entiende el uso de la lógica para representar problemas y métodos de resolución de problemas (en forma de hechos y reglas) junto con la utilización de procedimientos automáticos de prueba para la solución efectiva de los problemas. Tal y como define Kowalski: *algoritmo = lógica + control.*

La programación lógica trabaja con relaciones más que con funciones. Un programa convencional distingue entre datos de entrada (argumentos) y datos de salida (resultados). El equivalente en Prolog es una relación entre todas las variables que intervienen, sin distinguir entre argumentos y resultados. En general, esa relación sirve para encontrar los valores posibles de cualquier variable a partir de los valores conocidos del resto.

La programación lógica –en particular, el *Prolog*– se utilizó, en sus orígenes, en el campo del procesamiento del lenguaje natural. Desde entonces, su uso se ha extendido a la construcción de sistemas expertos, búsquedas en bases de datos, diseño de compiladores, especificación de algoritmos, etc.

TEMARIO DETALLADO

Unidad Didáctica I: Programación funcional

TEMA 1. CONCEPTOS FUNDAMENTALES

1.1. Lenguajes funcionales. Lenguajes. Funciones.

1.2. Intérpretes. El intérprete De Hugs/Gofer. Computar expresiones. Definir funciones. Comandos del intérprete.

1.3. Funciones estándar. Funciones primitivas y predifinidas. Nombres de funciones y operadores. Funciones sobre números. Funciones booleanas. Funciones sobre listas. Funciones sobre funciones.

1.4. Definición de funciones. Por combinación. Por distinción de casos. Por reconocimiento de patrones. Por recursión o inducción. Tipografía y comentarios.

1.5. Tipos. Errores. Expresiones de asignación de tipo. Polimorfismo. Funciones con más de un argumento. Sobrecarga.

TEMA 2. FUNCIONES Y NÚMEROS

2.1. Operadores. Operadores como funciones y viceversa. Prioridades. Asociación. Definición de operadores.

2.2. Currificación. Parametrización parcial. Paréntesis. Sección de operadores.

2.3. Funciones como argumentos. Funciones sobre listas. Iteración. Composición.

2.4. Funciones numéricas.

TEMA 3. ESTRUCTURAS DE DATOS CARACTERÍSTICAS

3.1. Listas. Estructura de una lista. Funciones sobre listas. Funciones de orden superior sobre listas. Ordenación de listas.

3.2. Listas especiales. Cadenas de caracteres. Funciones sobre cadenas de caracteres. Listas infinitas. Evaluación perezosa. Funciones sobre listas infinitas.

3.3. Tuplas. Uso de tuplas. Definición de tipos. Números racionales. Listas y tuplas. Tuplas y currificación.

3.4. Árboles. Definiciones de datos. Árboles de búsqueda. Uso especial de definiciones de datos.

Unidad Didáctica II: Programación lógica

TEMA 1. CÁLCULOS CON RELACIONES

Relaciones. Reglas y hechos. Consultas. Programación lógica.

TEMA 2. INTRODUCCIÓN A PROLOG

Términos. Interacción con Prolog. Consultas de existencia. Hechos y reglas universales. La negación como fracaso. Unificación. Aritmética.

TEMA 3. ESTRUCTURAS DE DATOS EN PROLOG

Listas. Términos como datos.

TEMA 4. TÉCNICAS DE PROGRAMACIÓN

Propone y verifica. Variables como lugares para acomodar datos. Listas de diferencias.

TEMA 5. CONTROL EN PROLOG

Unificación y substituciones. Aplicación de una regla a una meta. Árboles de búsqueda en Prolog. El orden de metas cambia las soluciones. El orden de las reglas afecta la búsqueda de soluciones. El problema de la aparición y la verificación.

TEMA 6. CORTES

Un corte como primera condición. El efecto de un corte. Aplicaciones de cortes en programación. La negación como fracaso.

EQUIPO DOCENTE

Nombre y Apellidos	JULIO ANTONIO GONZALO ARROYO
Correo Electrónico	julio@lsi.uned.es
Teléfono	91398-7922
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS

BIBLIOGRAFÍA BÁSICA

La primera unidad didáctica está dedicada a la programación funcional. Como la programación lógica se estudia también en la asignatura “Programación orientada a la Inteligencia Artificial”, en esta asignatura se pone un mayor énfasis en la programación funcional, y la práctica está relacionada con esta parte de la asignatura. El programa se ajusta a los tres primeros capítulos del libro de J. Fokker: *Programación funcional, Universidad de Utrecht, 1995*, que les será proporcionado a los alumnos. Como texto complementario al libro anterior es recomendable utilizar el manual de Hugs.

La segunda unidad didáctica está dedicada a la programación lógica y, en particular, al lenguaje Prolog. Para el estudio de esta unidad se proporcionará a los alumnos material a través del entorno virtual de la asignatura. No obstante, esta parte del programa coincide con el capítulo 8 del libro de R. Sethi: *Lenguajes de programación: conceptos y constructores, Addison-Wesley, 1992*.

Cualquier material adicional se encontrará en:
http://www.lsi.uned.es/pd
o accediendo al entorno virtual de la asignatura.

BIBLIOGRAFÍA COMPLEMENTARIA

SISTEMA DE EVALUACIÓN

Para aprobar la asignatura habrá que realizar una Práctica además de la prueba presencial y aprobar ambas por separado.

PRÁCTICA

Habrá una prueba práctica de caracter obligatorio que consistirá en realizar diversos programas utilizando el lenguaje funcional *Hugs*, en los que se pondrán en juego los conocimientos adquiridos en la asignatura.

Todo el material necesario para la realización de las prácticas se pondrá a disposición del alumno a través de los Cursos Virtuales de la asignatura, medio que utilizará el Equipo Docente para monitorizar la realización de las prácticas por parte de los alumnos.

Para aprobar la asignatura será **requisito imprescindible** aprobar la práctica en el mismo curso y con anterioridad a la prueba presencial. Aquellos alumnos que deseen concurrir a la Convocatoria Extraordinaria de fin de Carrera, deberán tener aprobadas las prácticas en el curso inmediatamente anterior.

ORIENTACIONES SOBRE LA PRUEBA PRESENCIAL

Consistirá en ejercicios y problemas. El peso de la Unidad Didáctica I (programación funcional) será siempre mayor que el de la programación lógica. No se permitirá el uso de material para realizar la prueba presencial.

HORARIO DE ATENCIÓN AL ESTUDIANTE

El horario de guardias será los jueves de 16 a 20 h.

MEDIOS DE CONTACTO CON EL EQUIPO DOCENTE

En orden de preferencia:

1. A través de los foros de los cursos virtuales
2. A través del correo electrónico de la asignatura: progde@lsi.uned.es
3. Teléfonos del Equipo Docente:

•Fernando López Ostenero: 91 398 77 93

•Julio Gonzalo Arroyo: 91 398 79 22

SERVIDOR WWW DE LA ASIGNATURA

http://www.lsi.uned.es/pd

IGUALDAD DE GÉNERO

En coherencia con el valor asumido de la igualdad de género, todas las denominaciones que en esta Guía hacen referencia a órganos de gobierno unipersonales, de representación, o miembros de la comunidad universitaria y se efectúan en género masculino, cuando no se hayan sustituido por términos genéricos, se entenderán hechas indistintamente en género femenino o masculino, según el sexo del titular que los desempeñe.