Programa de Asignatura

Ingeniería en Computación - In.Co.

Nombre de la Asignatura	Introducción a la Programación Genérica.
Créditos	10.
Objetivo de la Asignatura	El objetivo de este curso es introducir los conceptos fundamentales relativos al diseño y transformación de programas funcionales sobre la base de una interpretación algebraica de tipos de datos. En particular, se muestra como dicho enfoque algebraico de tipos permite en forma natural la construcción de programas genéricos (también llamados politípicos), que son programas paramétricos en la signatura (estructura) de los tipos de datos que manipulan.
Metodología de enseñanza	El curso tendrá una duración de 10 semanas. Semanalmente se dictarán 4 horas de clase teórico-prácticas en las cuales se presentarán los conceptos y técnicas fundamentales. En forma complementaria el estudiante deberá profundizar los temas dados en clase mediante el estudio de la bibliografía sugerida, para lo cual se estima una dedicación promedio de 4 horas semanales. Más las dedicadas a ejercicios obligatorios.
Temario	1- Introducción y Motivación. 2- Conceptos preliminares: Tipos de datos básicos (producto, suma) y sus leyes algebraicas. El concepto de functor, functores polinomiales. Transformaciones naturales. 3- Tipos inductivos: Interpretación categórica de tipos de datos. Functores como signaturas, álgebras de functores, álgebras iniciales. El operador fold y sus leyes algebraicas. Functores de tipo, functores regulares. 4- Funciones politípicas. 5- Tipos co-inductivos: Co-álgebras; co-álgebras finales. El operador unfold y sus leyes algebraicas. Functores de tipo. 6- Interpretación de tipos de datos como dominios: El operador hylomorfismo y sus leyes algebraicas.
Bibliografía	Algebra of Programming, Richard Bird y Oege de Moor, Prentice Hall, 1997. ISBN 013507245-X Introduction to Functional Programming using Haskell, Richard Bird, Prentice Hall, 1998. ISBN 0134843460 Generic Programming - An Introduction -, Roland Backhouse, Patrick

Jansson, Johan Jeuring y Lambert Meertens, Advanced Functional Programming, Lecture Notes in Computer Science, 1608, Springer-Verlag, 1998.

Polytypic Programming, Johan Jeuring y Patrick Jansson, Advanced Functional Programming, Lecture Notes in Computer Science, 1129, Springer-Verlag, 1996.

En el curso se va a hacer referencia a otras publicaciones técnicas relacionadas.

Conocimientos previos exigidos y recomendados

Es deseable que el estudiante tenga también cierta familiaridad con nociones básicas de semántica formal de lenguajes de programación, en especial, semántica denotacional.

Anexo:

1) Un Cronograma tentativo.

Semana 1
- Motivación
- El operador foldr.
- Definición de categoría,.
- Functores y transformaciones naturales.

Semana 2
- Objeto final (terminal) y objeto inicial.
- Producto, leyes algebraicas, functor producto.
- Coproducto, leyes algebraicas, functor suma.
- Functor composición.
- Functores polinomiales.

Semana 3
- Algebras, signaturas, homomorfismos, el álgebra de términos, el concepto de álgebra inicial.
- Signaturas como functores, interpretación categórica de ecuaciones recursivas de tipo.
- F-álgebras, F-homomorfismos, categoría de las F-álgebras.
- Punto fijo de un functor, F-álgebras iniciales.
- El operador fold y su propiedad universal, ley de fusión.

Semana 4
- Ley de deforestación y ley banana split (tupling).
- Bifunctores, tipos de datos paramétricos, functores de tipo.
- Ley de fusión entre map y fold.

Semana 5
- Funciones politípicas. Prueba de propiedades sobre funciones politípicas.

Semana 6
- Tipos coinductivos.
- El operador unfold y su propiedad universal.
- Leyes de fusión.
- DFS y BFS sobre árboles.

Semana 7
- El operador hylomorfismo.
- Leyes de fusión.
- Aplicaciones.

Semana 8
- Lenguajes para programación genérica.
- Temas de investigación en el área.

Semana 9
- Presentación de artículos por parte de los estudiantes.

Semana 10

Presentación de artículos por parte de los estudiantes.

2) Modalidad del curso y procedimiento de evaluación.

A lo largo del curso se asignarán ejercicios obligatorios para realizar en forma personal con el objetivo de luego ser entregados para su corrección. Se estima que, en su totalidad, el práctico de la materia requiere una dedicación de 60 horas.

Complementariamente, se requiere a cada estudiante la lectura y exposición de un artículo técnico (extraído de revistas y/o conferencias) que trate temas relacionados con el curso. Asimismo se exigirá entregar un resumen de dicho artículo en el cual se describan los aspectos relevantes del mismo.

El curso consistirá también de un laboratotio donde se practicará el uso de lenguajes experimentales que permiten escribir programas genéricos, como por ejemplo, PolyP o Generic Haskell.

3) Materia.

Programación.

4) Previaturas.

Tener aprobada la Electiva de Introducción a la Programación Funcional (plan 87).
Tener aprobada la asignatura Introducción a la Programación Funcional (plan 97).

5) Cupo

Para los estudiantes de la carrera un cupo de 30.

Aprobado por Res. del Consejo de fecha 22.3.2001 ,Exp.060120-000339-01