

PROGRAMA de Características de Lenguajes de Programación

Carreras: Licenciatura en Informática y Tecnicatura en Programación Informática

Asignatura: Características de Lenguajes de Programación

Núcleo al que pertenece: Avanzado

Profesores: Alejandro Díaz-Caro y Lucas Rafael Romero

Asignaturas Correlativas: Lógica y Programación

Objetivos:

La materia está planteada para el noveno semestre, cuando ya se han consolidado las habilidades de desarrollo de software. A esas habilidades les falta la comprensión de los fundamentos de los diferentes paradigmas, así como las diferentes técnicas que utilizan los lenguajes de programación modernos para lidiar con cada característica: inferencia de tipos, pasaje de parámetros, administración de memoria, etc. Es por eso que el presente curso, a esta altura de la carrera, propone estudiar las diferentes características de los lenguajes, otorgando un panorama general.

Contenidos mínimos:

- Lenguajes según su modelo de cómputo:
 - Modelos de cómputo imperativo, funcional, objetos, lógico
 - Realización de estructuras de datos en los diferentes paradigmas
- Lenguajes según sus características:
 - Lenguajes tipados y no tipados. Sistemas de tipos
 - Mecanismos de binding (estático y dinámico)
 - Mecanismos de pasaje de parámetros (valor, referencia, nombre, otros)
 - Formas de llevar a cabo la ejecución (compilación, interpretación, máquinas virtuales)
 - Formas de administración de memoria (explícita y garbage collection)
- Lenguajes según su propósito

- Lenguajes de propósitos generales
- Lenguajes de dominio específico
- Lenguajes de scripting
- Aptitudes de diferentes lenguajes para diferentes tareas (claridad, eficiencia, modificabilidad, etc.)
- Lenguajes según la forma de asignarles significado
 - Herramientas de asignación de significado (semánticas operacional, denotacional, axiomática)
 - Casos específicos de semántica operacional, ilustrando modelos de cómputo y características

Carga horaria semanal: 4 hs

Programa analítico:

Unidad 1: PCF no tipado

Primeras definiciones. Gramática de PCF. Semántica operacional. No terminación. Captura de variables

Unidad 2: Estrategias de reducción

Primeras definiciones. Reducción débil. Call-by-name. Call-by-value

Unidad 3: PCF tipado

Introducción. Gramática de PCF tipado. La relación de tipado

Unidad 4: Inferencia de tipos simples

Introducción. Algoritmo de Hindley. Algoritmo de unificación de Robinson

Unidad 5: Polimorfismo

Introducción. Tipos polimórficos. Sistema dirigido por sintaxis de Hindley-Milner

Unidad 6: Formas de llevar a cabo la ejecución

Interpretación: Interpretación en CBN e Interpretación en CBV. Compilación: Una máquina abstracta para PCF

Unidad 7: Registros y Objetos

Registros: Campos etiquetados y Extensión de PCF con registros. Objetos: Los métodos y los campos funcionales, ¿Qué es "self"? y PCF con objetos

Unidad 8: Semántica denotacional

Primeras definiciones. La semántica denotacional de PCF tipado

Unidad 9: Dos demostraciones importantes

Subject reduction. Normalización fuerte

Unidad 10: Administración de memoria

Diferentes tipos de memoria (de código, de datos, de pila, dinámica). Manejo de memoria explícita. Manejo automático de memoria

Unidad 11: Introducción a la lógica lineal

Unidad 12: Lenguajes para diferentes propósitos

Ejemplos de papers recientes.

Bibliografía obligatoria:

1. G. Dowek y J.-J. Lévy "Introduction to the theory of programming languages", Springer, 2011.
2. D. P. Friedman, M. Wand y C. T. Hayner "Essentials of programming languages", MIT Press, 2011.
3. F. Pfenning "Linear Logic", Carnegie Mellon University, 2002.
4. A. Díaz-Caro y P. Martínez López "Isomorphisms considered as equalities: Projecting functions and enhancing partial application through an implementation of λ ", ACM Proceedings of IFL'15(9), 2015.
5. A. Díaz-Caro y G. Dowek "Typing quantum superpositions and measurement", LNCS 10687:281-293, 2017.

Bibliografía de consulta:

1. G. Dowek "Principles of programming languages", Springer, 2009.

Organización de las clases

Las clases serán en modalidades teórica y práctica. En las clases teóricas se explicarán los conceptos fundamentales relacionados a las características de los lenguajes de programación, utilizando como base a lo largo de toda la materia el cálculo lambda.

Las clases teóricas se reforzarán con un material de lectura que se dará al finalizar cada clase y un ejercicio que deberán realizar y entregar por email antes de la siguiente clase. El objetivo de ese ejercicio no es de evaluación sino de asentar los conceptos trabajados en clase. Por lo tanto, sólo se exigirá la entrega, estén o no resueltos correctamente, y se hará una devolución por email, que luego podrá ser expandida en la clase práctica correspondiente.

Descripción de las actividades prácticas

Las clases prácticas consisten en la resolución de ejercicios que serán entregados al inicio del semestre. Durante ellas se dará tiempo para la realización individual de los ejercicios, consultando cuando sea necesario, y se desarrollarán en el pizarrón

algunos de ellos (luego de que hayan tenido el debido tiempo para razonar sobre ellos).

Unidad 1: PCF no tipado

El objetivo principal de esta práctica es lograr familiaridad con la notación y la sintaxis de PCF mediante la escritura y evaluación de expresiones en el lenguaje PCF no tipado así como a definiciones de importancia en el resto de la materia (normalización, confluencia, etc.). Además se extiende la sintaxis del lenguaje para admitir estructuras con mayor complejidad como pares, listas y constantes booleanas.

Unidad 2: Estrategias de reducción

En esta práctica se espera que arriben a la comprensión, mediante casos particulares, de los diferentes tipos de reducciones (fuerte, débil) y las estrategias (call-by-name, call-by-value). Por último, se extienden las estrategias de reducción para las estructuras y elementos adicionales dados en la práctica anterior.

Unidad 3: PCF tipado

Las actividades prácticas de esta unidad profundizan y ejercitan las nociones de extensión de un lenguaje para incluir definiciones de tipos, con especial énfasis en la definición de una relación de tipado. Se presentan expresiones para tipar y se espera que puedan distinguir aquellas que poseen tipo de las que no pueden tenerlo, y dar una derivación del tipo de las primeras.

Unidad 4: Inferencia de tipos simples

En esta sección se complementa el desarrollo teórico con ejemplos de aplicación de los algoritmos de inferencia de tipos. Un gran número de incisos prácticos consisten en desarrollar ambos algoritmos sobre expresiones del lenguaje. Adicionalmente, se trabaja con las extensiones del lenguaje heredadas de las prácticas anteriores (pares, listas, árboles, etc.)

Unidad 5: Polimorfismo

Para esta unidad, las actividades prácticas buscarán ejercitar las reglas de derivación de tipos polimórficos. Se pone especial énfasis en el ejercicio de dar los tipos más generales de las expresiones propuestas. Se presentan además las extensiones de las reglas de derivación para el lenguaje original enriquecido con estructuras más complejas. Por último, se aplica el sistema dirigido por sintaxis de Hindley-Milner en algunas expresiones.

Unidad 6: Formas de llevar a cabo la ejecución

El objetivo de esta práctica es lograr entender las construcciones abstractas para la interpretación y compilación, mediante su ejecución en casos concretos. Se ponen en relieve las diferencias y similitudes de interpretación en call-by-name y call-by-value trabajando ambas formas en las mismas expresiones. Por último, se extiende el intérprete a pares, listas y booleanos.

Unidad 7: Registros y Objetos

Las actividades prácticas para registros y objetos consisten en la definición de instancias y objetos concretos mediante la sintaxis del lenguaje (por ejemplo, números complejos).

Unidad 8: Semántica denotacional

Los ejercicios propuestos en esta sección consisten en la evaluación semántica de expresiones del lenguaje. Se pone especial atención en la semántica de expresiones que involucren el operador de punto fijo. Por último, como parte de la práctica se pide demostrar el teorema de soundness (con ayuda en los pasos intermedios necesarios).

Unidad 9: Dos demostraciones importantes: subject reduction y normalización fuerte.

En esta unidad se pide que puedan construir algunas demostraciones simples que son dejadas como ejercicio en el desarrollo teórico. Además, se pide extender las dos demostraciones (extendiendo sus lemas) para el caso de estructuras más complejas del lenguaje (pares, listas, etc.).

Unidad 10: Administración de memoria

Para esta unidad se les requiere la elaboración de un trabajo práctico con formato de monografía breve sobre la administración de memoria en sistemas operativos populares. La fecha de entrega del trabajo se acuerda al presentar las consignas y usualmente es de quince días.

Unidad 11: Introducción a la lógica lineal

Para la última actividad práctica, se dan expresiones concretas en lógica lineal y se pide construir árboles de derivación así como extender las reglas dadas en la teoría para pares, listas y constantes booleanas.

Unidad 12: Lenguajes para diferentes propósitos

En esta unidad se les pide leer un paper publicado recientemente en alguna conferencia internacional importante (POPL, LICS), sobre algún lenguaje de propósito específico.

Modalidad de evaluación

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18).

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial.

CRONOGRAMA TENTATIVO

| Semana | Tema/unidad | Actividad* | | | | Evaluación |
|--------|---|------------|-----------|------|-------------------|------------|
| | | Teórico | Práctico | | | |
| | | | Res Prob. | Lab. | Otros Especificar | |
| 1 | PCF no tipado (Unidades 1) | X | X | | | |
| 2 | Estrategias de reducción (Unidad 2) | X | X | | | |
| 3 | PCF tipado (Unidad 3) | X | X | | | |
| 4 | Inferencia de tipos (Unidad 4) | X | X | | | |
| 5 | Polimorfismo (Unidad 5) | X | X | | | |
| 6 | Interpretación (Unidad 6) | X | X | | | |
| 7 | Compilación y máquinas virtuales (Unidad 6, continuación) | X | X | | | |
| 8 | Repaso | | | | Consulta | |
| 9 | 1er Parcial | | | | | X |
| 10 | Registros y objetos (Unidad 7) | X | X | | | |
| 11 | Semántica de Lenguajes (Unidad 8) | X | X | | | |
| 12 | Subject reduction y strong normalisation (Unidad 9) | X | X | | | |
| 13 | Formas de administración de memoria (Unidad 10) | X | | X | | |
| 14 | Introducción a la lógica lineal (Unidad 11) | X | | X | | |
| 15 | Lenguajes según su propósito (Unidad 12) | X | | X | | |
| 16 | Repaso | | | | Consulta | |
| 17 | 2do Parcial | | | | | X |
| 18 | Recuperatorio | | | | | X |
| 19 | Integrador | | | | | X |

*INDIQUE CON UNA CRUZ LA MODALIDAD