

Planificación de Compiladores



Código/s: R-422

Identificación y características del Espacio Curricular

| | | | |
|----------------------|--|---------------------|----------------------------|
| Carrera/s: | Licenciatura en Ciencias de la Computación | | |
| Plan de Estudios: | 2010, TO2024 | Carácter: | Obligatoria |
| Bloque/Campo: | | Área: | Algoritmos y Lenguajes |
| Régimen de cursado: | Cuatrimestral | | |
| Cuatrimestre: | 8º [LCC], 8º [LCC] | | |
| Carga horaria: | 120 hs. / 8 hs. semanales | Formato curricular: | Asignatura |
| Escuela: | Ciencias Exactas y Naturales | Departamento: | Ciencias de la Computación |
| Docente responsable: | JASKELIOFF, Mauro | | |

Programa Sintético

Conceptos de intérpretes y compiladores. Análisis léxico y parsing. Jerarquía de Chomsky. Gramáticas libre de contexto. Herramientas. Sintaxis abstracta. Análisis semántico. Evaluación de expresiones tipadas y/o mutuamente recursivas. Implementación de variables locales. Código intermedio. Traducción de AST a IRT. Bloques básicos. Árboles canónicos. Serialización. Selección de instrucciones. Maximal Munch y BURS. Análisis de tiempo de vida. Interferencias. Flujo de datos. Uso de heurística en algoritmos. Alojamiento de registros: algoritmos de Sethi-Ullman y de Chaitin. Emisión de Assembler. Run Time.

Espacios Curriculares Relacionados

| | |
|---------------------------|---|
| Previos Aprobados: | R-213 - Lenguajes Formales y Computabilidad, R-312 - Estructuras de Datos y Algoritmos II, R-322 - Análisis de Lenguajes de Programación, R-314 - Examen de Suficiencia de Inglés |
| Simultaneos Recomendados: | R-423 - Complementos de Matemática II, R-421 - Ingeniería del Software II |
| Posteriores: | |

Vigencia desde 2024

Firma Profesor

Fecha

Firma Aprob. Escuela

Fecha

Con el aval del Consejo Asesor:

Fundamentación

La definición de lenguajes de programación y su implementación con intérpretes y compiladores es una actividad central en el ejercicio de las Ciencias de la Computación. Esta materia presenta los aspectos generales y concretos de este campo. El enfoque, que prioriza este objetivo, es que el alumno lleve adelante en forma colaborativa un proyecto en el que se construye un compilador para un lenguaje no trivial.

Se encuentra en el cuarto año, segundo cuatrimestre, y en la misma se capitalizan, se potencian, y se ponen en práctica en un proyecto de magnitud considerable conocimientos adquiridos en materias anteriores como ser Lenguajes Formales y Computabilidad, Estructuras de Datos y Algoritmos I y II, Arquitectura del Computador, Análisis de Lenguajes de Programación.

Resultados del aprendizaje

Al finalizar el cursado los/las estudiantes serán capaces de:

RA1 Identificar la problemática del desarrollo de intérpretes y compiladores, elegir las técnicas adecuadas para resolverla y aplicarlas a un problema concreto.

RA2 Implementar intérpretes y compiladores de lenguajes de programación.

RA3 Gestionar, planificar y ejecutar un proyecto de informática no trivial.

RA4 Asumir responsabilidades dentro de un equipo de trabajo.

RA5 Trabajar en equipo utilizando en forma efectiva herramientas para el desarrollo colaborativo de software.

Competencias / Ejes transversales y Resultados del Aprendizaje

| Competencia/Eje transversal al que tributa | Nivel | Resultados del Aprendizaje |
|--|-------|----------------------------|
| CGT1-Identificación, formulación y resolución de problemas de informática | Alto | RA1 RA2 |
| CGT3-Gestión, planificación, ejecución y control de proyectos de informática | Alto | RA3 |
| CGS1-Fundamentos para el desempeño en equipos de trabajo | Alto | RA4 RA5 |

Programa Analítico

Unidad 1

- 1.1 Definición y objeto de intérpretes y compiladores.
- 1.2 Organización por módulos.
- 1.3 Organización del proyecto y metodología de trabajo.
- 1.4 Herramientas de construcción, testeado y versionado.

Unidad 2

- 2.1 Lenguaje objeto. Sintaxis concreta y abstracta.
- 2.2 Sistemas de tipos.
- 2.3 Uso del operador de punto fijo.

Unidad 3

- 3.1 Sintaxis Abstracta.
- 3.2 Representación de variables ligadas. Índices de De Bruijn y representación localmente sin nombres.
- 3.3 Azúcar sintáctico y expresividad.

Unidad 4

- 4.1 Análisis léxico y Parsing.
- 4.2 Herramientas.

Unidad 5

- 5.1 Máquinas abstractas.
- 5.2 Semántica operacional estructural.
- 5.3 Máquina CK.
- 5.4 Máquina CEK.
- 5.5 Otras máquinas abstractas.

Unidad 6

- 6.1 Bytecode y máquinas virtuales.
- 6.2 Diseño e implementación de una máquina virtual.
- 6.3 Optimización de llamadas de cola.

Unidad 7

- 7.1 Optimización de código.
- 7.2 Recursión de Cola
- 7.3 Cálculo estático de constantes.
- 7.4 Eliminación de código muerto.
- 7.5 Expansión en línea.
- 7.6 Eliminación de subexpresiones comunes.

Unidad 8

- 8.1 Conversión de clausuras.
- 8.2 Representación de clausuras.
- 8.3 Clausuras de funciones recursivas.
- 8.4 Algoritmos de conversión de clausuras.

Unidad 9

- 9.1 Bloques básicos y trazas.
- 9.2 Árboles canónicos.
- 9.3 Transformaciones de saltos condicionales.
- 9.4 Serialización y determinación de bloques básicos.
- 9.5 Posibles trazas.

Unidad 10

- 10.1 Emisión de código.
- 10.2 Paquetes y rutinas de Runtime.
- 10.3 Técnicas generales y específicas.

Unidad 11

- 11.1 Recolección de basura.
- 11.2 Conteo de referencias.
- 11.3 Marcar y barrer.
- 11.4 Recolección con copia.
- 11.5 Recolector de Boehm-Demers-Weiser.

Modalidades de enseñanza

El formato curricular adoptado es el de aprendizaje basado en un proyecto de desarrollo, realizado en el formato de trabajo grupal. Se dan clases teóricas motivadas por la necesidad de avanzar en el proyecto.

Esta modalidad se relaciona con el resultado de aprendizaje RA3. El proyecto consiste en el desarrollo de diferentes formas de evaluación de programas de un lenguaje de programación determinado, lo cual se relaciona de forma directa con los resultados de aprendizaje RA1 y RA2. La modalidad de trabajo grupal, y la metodología propuesta de desarrollo del proyecto se relacionan con los resultados de aprendizaje RA4 y RA5.

Recursos

Se utiliza un laboratorio informático para el desarrollo de la actividad. Para las clases teóricas se utiliza un proyector. Para el desarrollo colaborativo se utiliza una plataforma abierta de control de versiones y de seguimiento de tareas.

Todo el material de estudio se sube al campus virtual (comunidades) provisto por la Universidad para que todos los alumnos tengan acceso al material. Para la comunicación se utiliza un canal de Zulip que corre en un servidor del Departamento de Ciencias de la Computación.

Actividades de Formación Práctica

| Nº | Título | Descripción |
|----|-------------------------------------|---|
| 1 | Lenguaje PCF | Introducción al lenguaje teórico PCF. Ejercicios en papel |
| A | Reunión inicial | Conformación de grupos de trabajo, participantes y roles. |
| 2 | Lenguaje FD4 | Introducción al proyecto y al lenguaje concreto FD4 (basado en PCF). Puesta en marcha del proyecto y ejercicios de familiarización con el mismo |
| 3 | Representación de variables ligadas | Representación de variables ligadas usando la representación locally nameless. Ejercicios en papel. |
| 4 | Azúcar sintáctico | Implementación de azúcar sintáctico. Extensión del parser, elaborador y pretty-printer |
| B | Reunión de Control | Evaluación de avance, metodología y trabajo grupal |
| 5 | Máquinas abstractas | Implementación de un backend sobre máquina abstracta CEK. |
| 6 | Máquinas virtuales | Implementación de compilación a aq bytecode y de máquina virtual. |
| 7 | Optimizando la máquina virtual | Implementación de optimización de recursión de cola (mas otras). |
| C | 2da Reunión de control | Evaluación de avance, metodología y trabajo grupal |
| 8 | Optimizaciones | Implementación de Optimizaciones generales (útiles para todos los backend) |
| 9 | Conversión de clausuras | Implementación de conversión de clausuras. Emisión de código C. |
| D | 3ra reunión de control | Evaluación de avance, metodología y trabajo grupal |
| 10 | Código IR de bajo nivel | Implementación de compilación a LLVM. |
| E | Entrega de proyecto | Se entrega un proyecto que pasa todos los tests provistos por la cátedra. |
| F | Defensa del proyecto. | Presentación de los resultados, problemas y soluciones. |

Evaluación

Se regulariza la asignatura habiendo aprobado la 1a y 2da reunión de control (B y C). Se aprueba aprobando al menos dos de las reuniones de control y entregado y defendido exitosamente el proyecto.

En la reunión inicial se definirán la composición y roles del equipo y el nombre del equipo, se hará una evaluación inicial de la capacidad de negociar roles y llegar a acuerdos. En las reuniones de control se evaluará el correcto desempeño del trabajo en grupo y la gestión y planificación del proyecto. La evaluación consta de dos partes: una autoevaluación realizada por los estudiantes, más un coloquio. Los resultados se anotarán en una planilla dónde se registran los siguientes indicadores: participación, colaboración, involucramiento, espíritu de equipo y capacidad de reflexionar sobre el funcionamiento.

Para aprobar y defender exitosamente el proyecto se definen requisitos mínimos de funcionamiento, más partes opcionales. Con los requisitos mínimos se aprueba el proyecto con nota mínima (6). La implementación exitosa de opcionales mejoran la nota final del proyecto. La defensa consiste en la presentación de las principales decisiones de diseño más un coloquio.

| Resultado de Aprendizaje | Actividades/Modalidad de Enseñanza | Modalidad de Evaluación |
|--------------------------|------------------------------------|---|
| RA1 | E F | (E) mediante test automatizados (F) Interactiva y cualitativa durante la defensa del proyecto. |
| RA2 | E F | (E) mediante test automatizados (F) Interactiva y cualitativa durante la defensa del proyecto. |
| RA3 | ABCDF | Interactiva y cualitativa durante las reuniones y la defensa del proyecto. |
| RA4 | ABCDF | Interactiva y cualitativa durante las reuniones y la defensa del proyecto. |
| RA5 | ABCDF | Interactiva y cualitativa durante las reuniones y la defensa del proyecto. |

Bibliografía básica

| Autores (Apellido, Inicial nombre) | Año de edición | Título de la obra | Editorial o Revista | Ejemplares disponibles o sitio web |
|------------------------------------|----------------|--------------------------------------|----------------------------|------------------------------------|
| Appel, A | 1998 | Modern Compiler Implementation in ML | Cambridge University Press | 1 |
| Pierce, Benjamin | 2002 | Types and Programming Languages | MIT Press | 2 |

Bibliografía complementaria

| Autores (Apellido, Inicial nombre) | Año de edición | Título de la obra | Editorial o Revista | Ejemplares disponibles o sitio web |
|------------------------------------|----------------|---|---------------------|------------------------------------|
| Peyton-Jones, S. | 1987 | The Implementation of Functional Programming Languages. | Prentice-Hall | 1 |

| | | | | |
|---------------------------------|------|---|---|---|
| Abelson, H. and G. Sussman. | 1996 | Structure and Interpretation of Programming Languages | MIT Press (2nd ed) | https://web.mit.edu/6.001/6.037/sicp.pdf |
| Felleisen, M. and Friedman, D.I | 1986 | Control Operators, the SECD Machine, and the Lambda-Calculus | Formal Description of Programming Concepts III, Elsevier Science Publishers B.V. (North-Holland). | https://www.iu.edu/search/ |
| Marlow, S. and Peyton-Jones, S. | 2006 | Making a fast curry: push/enter vs. eval/apply for higher-order languages | | https://www.microsoft.com/en-us/research/publication/make-fast-curry-pusher-vs-evalapply/ |
| Leroy, X. | 1990 | The Zinc Abstract Machine | | https://caml.inria.fr/pub/papers/xleroy-zinc.pdf |

Distribución de la carga horaria

Presenciales

| | | |
|--------------|---|----------------|
| Teóricas | | 52 Hs. |
| Prácticas | Formación Experimental | |
| | Resolución de Problemas vinculados a la Profesión | 20 Hs. |
| | Resolución de Problemas y Ejercicios | |
| | Actividades de Proyecto y Diseño | 40 Hs. |
| | Formación en la Práctica Profesional | |
| Evaluaciones | | 8 Hs. |
| Total | | 120 Hs. |

Dedicadas por el alumno fuera de clase

| | | |
|--------------|---|---------------|
| | Preparación Teórica | 10 Hs. |
| | Preparación Práctica | 70 Hs. |
| | Elaboración y redacción de informes, trabajos, presentaciones, etc. | 10 Hs. |
| Total | | 90 Hs. |

Cronograma de actividades

| Semana | Unidad | Tema | Actividad |
|--------|--------|---|-----------|
| 1 | 1 | Lenguaje PCF | 1 |
| 2 | 2 | Lenguaje FD4. Reunión inicial de comienzo del proyecto. | 2 A |
| 3 | 3 | Representación de variables ligadas | 3 |
| 4 | 4 | Azúcar sintáctico | 4 |
| 5 | 5 | Máquinas abstractas. Reunión de Control 1 | 5 B |
| 6 | 6 | Máquinas virtuales | 6 |
| 7 | | Optimizando la máquina virtual | 7 |
| 8 | | Reunión de Control 2 | C |

| | | | |
|----|-----|-------------------------|----|
| 9 | 7.3 | Optimizaciones | 8 |
| 10 | 7.5 | Optimizaciones | 8 |
| 11 | 8 | Conversión de clausuras | 9 |
| 12 | 9 | Reunión de Control 3 | D |
| 13 | 10 | Código IR de bajo nivel | 10 |
| 14 | 11 | Código IR de bajo nivel | 10 |
| 15 | | Entrega del proyecto | E |