



PROGRAMA ANALÍTICO DE ASIGNATURA

DEPARTAMENTO: Ingeniería en Sistemas de Información

CARRERA: Ingeniería en Sistemas de Información

NOMBRE DE LA ACTIVIDAD CURRICULAR: Sintaxis y Semántica de los Lenguajes

Año Académico: Plan 2023

Área: Programación

Bloque: Tecnologías Básicas

Nivel: 2º

Tipo: Obligatoria

Modalidad: Cuatrimestral

Cargas horarias totales:

<i>Horas reloj</i>	<i>Horas cátedra</i>	<i>Horas cátedra semanales</i>
96	128	8

OBJETIVOS

- Comprender la sintaxis y semántica de los lenguajes de programación
- Comprender los fundamentos de los lenguajes formales, gramáticas y autómatas
- Emplear conceptos y procedimientos de las gramáticas libres de contexto y gramáticas regulares en la especificación de la sintaxis de los lenguajes de programación
- Diferenciar los procesos de traducción de los lenguajes

CONTENIDOS

Contenidos mínimos

- Gramática y Lenguajes Formales
- Autómatas Finitos. Expresiones Regulares
- Gramáticas Independientes del Contexto
- Autómatas con Pila y Máquinas de Turing
- Análisis Léxico, Sintáctico y Semántico
- Traductores
- Proceso de Traducción

Contenidos analíticos



Parte I: Introducción a C y a Máquinas de Estado

Unidad 1: Lenguaje C

Historia y Evolución. Filosofía de Diseño y Necesidad que desea satisfacer. Proceso de Estandarización de LP y su Importancia. Biblioteca Estándar de C Uso actual. Relación con otros LP. Relación con C++. Principales Implementaciones: msvc, llvm, clang, gcc, Compiler Explorer.

Unidad 2: Conceptos Fundamentales y su Implementación en los LP

Símbolo, Caracteres, y Alfabeto: Operaciones de caracteres, Implementaciones, En C, En otros LP, ASCII, Unicode, UTF-8.

Cadena (String): Operaciones de cadena, Implementaciones, En C, En otros LP, Mutabilidad de longitud y contenido, Implementación en C.

Lenguaje y palabras: Clasificaciones, Naturales y Formales, Finitos e Infinitos, Jerarquía de Chomsky (LF), Lenguajes Regulares y Lenguajes No Regulares, Maneras de definir un lenguaje, Operaciones de lenguajes. Lenguaje Universal.

Abstracciones de C y otros LP para implementar los conceptos fundamentales: Valores: Tipos, Valores, y Variables, Arreglos, Cadenas, Flujos (Streams). Acciones: Expresiones, Estructuras de Control, Funciones.

Unidad 3: Jerarquía de Chomsky Lenguaje, Autómata y Gramática: LR ==> AF y GR / LIC ==> APD y GIC.

Gramáticas: GF, GIC, GR, el proceso de derivación por izquierda y por derecha.

Autómatas (Máquinas Reconocedoras): Máquinas de Estado, Autómata Finito, Push Down Autómata: PDA Determinístico y su definición formal. Aplicación en el reconocimiento de Lenguajes Independientes del Contexto, Máquina de Turing, Jerarquía de Chomsky (Autómata), Máquinas Accionadoras, Máquina de Moore, Máquina de Mealy.

Expresiones Regulares: Aplicaciones en Validaciones y Análisis Léxico. Ejemplos con grep o algún LP que las incorpore.

Logros Pedagógicos: Comprender el propósito y filosofía del lenguaje C para poder describir su impacto en la industria. Aplicar los conceptos de lenguajes formales para solucionar problemas de procesamiento de texto.

Parte II: Constructos del Lenguaje: Expresiones, Declaraciones, Sentencias y Estructura de Programa

Unidad 4: (Cp 2) Tipos, Operadores, y Expresiones

Tipo, precedencia, asociatividad, orden de evaluación. Declaraciones. Alcance. Incremento. Token y Evaluación de Lexema. (Cp3) Control de Flujo. Operaciones con cadenas. ciclos, saltos, selecciones.

Unidad 5: Especificación de LP Nivel Léxico

Lexemas. Tokens: Tipo y Valor. Representación. Categorías Léxicas. Identificadores: comunes, keywords, Palabras Reservadas, Identificadores Predefinidos. Literales: Numéricos: Enteros,



Caracteres, Flotantes. Sufijos y prefijos. Cadenas. Punctuators. Errores léxicos. Notaciones: Regex, GR, LN. Calculadora rudimentaria: Tokenizador de números.

Unidad 6: (Cp5) Funciones y Estructura de Programa.

Compilación Separada y Módulos. Calculadora polaca inversa: Tokenizador de operadores and operandos.

Logros Pedagógicos: Aplicar las abstracciones de los lenguajes de programación para la construcción de analizadores léxicos.

Parte III: Lenguajes de Programación y Traductores

Unidad 7: Traductores

Intérpretes. Compiladores. Assemblers (Ensambladores). Virtual Machines. Transpiladores. Tiempos. Ambiente de Ejecución. Ambiente de Traducción. Trabajo: Fases del ambiente de ejecución.

Unidad 8: Compilación separada y módulos. Visibilidad y Vinculación

Encapsulamiento. Headers. Biblioteca Estándar y Otras Bibliotecas. Trabajo: Automatización de compilación con make.

Unidad 9: Lenguajes de Programación

Propósito de Lenguajes de Alto Nivel. Clasificaciones (con programas ejemplo): Tipado o no tipado, Por tipo de Paradigma, Por Nivel de Abstracción, Compilado vs Interpretado, General o Específico, Garbage Collected. Árbol Genealógico. Lenguajes de Bajo Nivel: Ensamblador (Assembler), Lenguaje Assembly, mnemónicos, opcode, código máquina, bytecode.

Logros Pedagógicos: Comprender los diferentes tipos de lenguajes de programación y su proceso de traducción para eficientizar el desarrollo de software.

Parte IV: Especificación e Implementación de Lenguajes de Programación

Unidad 10: Introducción a la Especificación de lenguajes de programación

Consideraciones. Desafíos. Componentes de la especificación. Tipos de Especificación. Ejemplos: Especificación de Pascal (Railroad Diagram), Especificación de Smalltalk (Postcard). C (Apéndice A y Manual de Referencia).

Unidad 11: Sintaxis

Definición. Notaciones: BNF, BNF extendido, GIC, Railroad. Caso de estudio: C y otro LP. Constructos Comunes a varios LP: Estructura de Programa. Expresiones: Expresiones Primarias, Árbol de Expresión, Precedencia de Operadores, Asociatividad. Sentencias: Simples, Compuestas. Declaraciones: Declaradores, Definiciones, Inicializaciones. Errores sintácticos.



Unidad 12: Introducción a Semántica

Abstracción en los LP. Especificación de la Semántica con LN y con LF. Ambiente de Traducción y sus fases: Ocho Fases, Preprocesamiento, Vinculación. Estructura de Programa: Unidades de Traducción, Comunicación entre Unidades de Traducción. Consideraciones: Set de Caracteres, Semántica de Caracteres, Límites, Señales e Interrupciones. Ambientes. Ambiente de Ejecución y su soporte: Inicialización antes del comienzo del programa de objetos con duración estática. Comienzo del programa: main, argv, argc. Ejecución del programa: Efectos de Lado, Puntos de Secuencia Biblioteca Estándar. Terminación del programa: return, exit. Semántica del Léxico. Errores semánticos. Comportamiento: Definido. No definido. No especificado. Comportamiento Cultural. Especificado por la implementación. Portabilidad.

Unidad 13: Tipos

Definición y restricción de operaciones. Jerarquías: Tipos de Objeto, Tipos de Funciones. Tipos Derivados: Arreglos, Punteros, Uniones y Estructuras, Funciones Clasificación: Simples vs Compuestos, Básicos vs Derivados, Built-in vs Definidos por el programador. Conversiones: Promociones de Tipo, Implícitas, Explícitas, operador de casting. Constructores de Tipo: Producto Cartesiano, Unión, Enumeración.

Representación: Representación en código fuente, Representación en memoria, Valores trampa. Valores y L-Values Objetos (bloques de memoria). Storage Duration (Duración de Almacenamiento): Static (Estática), Thread, Automática: Stack, Uso actual de auto, Heap: Allocated ("Alocada"), RAII vs Garbage Collector (Recolector de Basura) Lifetime (Tiempo de Vida). Tipo Abstracto de Datos: Definición. Especificación: Sintaxis y Semántica. Implementación: Pública, Privada, Pruebas

Unidad 14: Declaraciones

Identificadores: Visibilidad o Alcance (scope): Bloque, Archivo, Prototipo de Función, Función. Vinculación: Interna, Externa, Ninguna. Namespaces: Colisiones, etiquetas, tags, miembros. Especificadores de Clase de Almacenamiento: register y uso actual. Calificadores de Tipo: volatile y iso actual. Const. Definiciones: Objetos, Inicialización, Funciones. Prototipos de Funciones. Inferencia de Tipo y auto de C++.

Unidad 15: Abstracción del control de flujo de Ejecución

Sentencias: Secuencia, Selección, Iteración, Saltos. Expresiones: Tipo de una expresión. Secuencia dentro de una Expresión: Precedencia de Operadores, Árbol de Expresión, Asociatividad Orden de Evaluación: Puntos de Secuencia, Cortocircuito, Estrategias de Evaluación. Efecto de Lado: Asignación, Operadores Matemáticos con y sin efecto de lado. Funciones: Pasaje de argumentos: Por Valor, Por Referencia, Referencia constante Retorno. Call Stack, Stack Frame, Recursividad. Pragmática: Uso de parámetros: in, out, inout

Unidad 16: Implementación de Lenguajes de Programación

Estructura de un Compilador. Bootstrapping de Compiladores. Diagramas Tombstone (T-Diagrams) Preprocesador. Mensajes de Error y Warnings.



Unidad 17: Front End

Analizador Léxico (Scanner): Manual, Ad-Hoc, Máquina de estado. Automático con flex. Ejemplos de la Industria.

Unidad 18 Analizador Sintáctico (Parser)

Tipos de Parsers, Manual, Tabla, Recursivo, Análisis Sintácticos Descendente y Ascendente. Las gramáticas adecuadas, Automático con bison, Árbol Sintáctico, Concreto, Abstracto, Ejemplos de la Industria.

Unidad 19 Analizador Semántico

Árbol Sintáctico Anotado. Tabla de Símbolos

Unidad 20 Back End

Optimizador, Generador de Código, Representaciones Intermedias, Código Objeto.

Logros Pedagógicos: Aplicar la especificación de sintaxis y la semántica de los lenguajes de programación para comprender analizar y comparar lenguajes existentes y crear nuevos. Aplicar las técnicas de diseño de compiladores para crear implementaciones de lenguajes simples.

BIBLIOGRAFÍA OBLIGATORIA

- Donnelly, Charles; Stallman, Richard . (2021). Bison: The Yacc-compatible Ed. Parser Generator. GNU.
- ISO/IEC. (2022). Programming languages — C, 5th Edition, C23. Open Standards.
- Kernighan, Brian; Ritchie, Dennis. (1991). *El Lenguaje de Programación C*. Ed. Prentice-Hall.
- Muchnik, J.; Díaz Bott, A. (2012). *Sintaxis y Semántica de los Lenguajes*. Ed. UTN FRBA CEIT.
- Paxson, V.; Estes, W.; Millaway, J. (2016). Lexical Analysis With Flex, for Flex 2.6.2.
- Scott, Michael L. (2016). *Programming Language Pragmatics, 4th Edition*. Morgan Kaufmann Publishers.
- Sola, J. (2021). *Identificadores: Alcance, Espacios de Nombre, Duración, y Enlace*. Ed. Construyendo Abstracciones.
- Sola, José María. (2021). *Niveles del Lenguaje: Léxico, Sintáctico, Semántico & Pragmático*. Ed. Construyendo Abstracciones.
- Sola, J. (2017). *Cadenas, Arreglos, Punteros, Pre, y Pos Incremento*. Ed. Construyendo Abstracciones.
- Sola, J. (2017). *Abstracciones, Listas Enlazadas, y For*. (2017). Construyendo Abstracciones.
- Stallman, R.; McGrath, R.; Smith; P. (2020). GNU Make: A Program for Directing Recompilation. GNU.



*Universidad Tecnológica Nacional
Facultad Regional Buenos Aires*

BIBLIOGRAFÍA COMPLEMENTARIA

- Cohen, Daniel (1986): *Introduction to Computer Theory*, Ed. Wiley.
- Gansner, Emden R.;Koutsofios, Eleftherios; North, Stephen. (2015). Graphviz.
- Holub, A. (1990): *Compiler Design in C*, Ed. Prentice-Hall.
- Stallman, R. M; the GCC Developer Community. (2022). Using the GNU Compiler Collection. GNU.
- The Clang Team. (2022). Clang Compiler User's Manual. Clang.
<https://clang.llvm.org/docs/UsersManual.html>
- Watson, Des (1989): *High-Level Languages and Their Compilers*, Ed. Addison-Wesley.

CORRELATIVAS

Para cursar y rendir

- Cursadas:
 - Lógica y Estructuras Discretas
 - Algoritmos y Estructuras de Datos
- Aprobadas: Ninguna