



DISCIPLINA: Compiladores

CÓDIGO: G03COMP0.01

VALIDADE: Início: **MARÇO/2022**

Término:

Carga Horária: Total: 60 horas/aula Semanal: 4 aulas Créditos: 4

Modalidade: Teórica

Classificação do Conteúdo pelas DCN: Profissional Integralização: Obrigatória

Ementa:

Introdução: A estrutura dos compiladores modernos: *front-end*, *middle-end*, *back-end*. Compiladores de um, dois e três passos. Análise léxica: Operações com expressões regulares. Reconhecimento de linguagens regulares com autômatos finitos. Construção de autômatos finitos deterministas a partir de expressões regulares. Geradores de varredores léxicos. Análise sintática: Sintaxe livre de contexto. Formas de derivação de strings e a árvore de sintaxe concreta. Precedência em expressões aritméticas. Eliminação de ambiguidade e de recursão à esquerda. Gramáticas LL(1) e LR(1). Derivação top-down. Derivação preditiva: fatoração à esquerda. Derivação recursiva: descendente e por tabelas de derivação. Recuperação de erros: o conjunto SYNCH. Gramáticas LL(K). Derivação bottom-up. Formas sentencias à esquerda e definição de manipuladores. Implementação por pilha: derivadores shift-reduce. Gramáticas LR(K). Construção de tabelas LR(0), SLR(1), LR(1), LALR(1). Análise semântica: Problemas sensíveis ao contexto. Ações semânticas em derivadores LL e LR. Gramáticas de atributos. Grafo de dependência de atributos. Estrutura e organização de tabelas de símbolos. Aninhamento léxico e regras de escopo. Descritores de tipos: formas de compatibilidade. Verificação e conversão de tipos em expressões. L-values e R-values. Representação intermediária para análise semântica: árvore de sintaxe abstrata. Ambientes de execução: Classes de armazenamento e acesso a dados não locais. Registros de ativação. Funções de mais alta ordem. Pilha de execução: criação e manipulação de registros de ativação. Geração de representação intermediária: Tipos de representação intermediária: árvores de sintaxe abstrata, grafo acíclico direcionado, grafo de controle do fluxo, código de três endereços. Regras semânticas para geração de código intermediário: atribuição e expressões, desvio de controle, declarações. Tradução em árvores de sintaxe abstrata. Reorganização do código intermediário: árvores canônicas, blocos básicos, aglomerados sequenciais. Geração de código de máquina para MIPS ou PENTIUM: Seleção de instruções. Análise de tempo de vida: grafos de fluxo do controle, grafos de interferência. Alocação de registradores: coloração de grafos, coalescência. Exemplo de otimização de laços.



Curso	Período	Eixo	Obrig.	Optativa
Engenharia de Computação	8º	Fundamentos de Engenharia de Computação	X	

Departamento/Coordenação: Departamento de Computação e Mecânica

INTERDISCIPLINARIDADES

Pré-requisitos	Código
Teoria da Computação	G03TCOM0.01
Co-requisitos	
Não há	

Objetivos: A disciplina deverá possibilitar ao estudante	
1	Conhecer os fundamentos de compiladores
2	Descrever as etapas de um compilador
3	Efetuar análise léxica de diversos tipos de linguagens e gramáticas
4	Construir autômatos finitos de acordo com uma especificação
5	Eliminar ambiguidades de gramáticas
6	Efetuar análises sintática e semântica
7	Implementar funcionalidades para geração de código intermediário e de máquina
8	Compreender e efetuar organização e gerenciamento de memória

Unidades de ensino		Carga-horária Horas/aula
1	INTRODUÇÃO 1.1 Processadores de linguagem 1.2 A estrutura de um compilador. 1.3 Expressões regulares, gramáticas e autômatos	4
2	ANÁLISE LÉXICA 2.1 O papel do analisador léxico 2.2 Tokens, lexemas e padrão de formação de tokens 2.3 Tabela de Símbolos 2.4 Expressões regulares e autômatos finitos 2.5 Implementação 2.6 Geradores de analisador léxico	8
3	ANÁLISE SINTÁTICA 3.1 O papel do analisador sintático 3.2 Gramáticas livres de contexto 3.3 Eliminação de ambiguidades 3.4 Análise sintática descendente 3.5 Análise sintática ascendente 3.6 Recuperação de erros 3.7 Implementação 3.8 Geradores de analisadores sintáticos	16

geralmente



4	TRADUÇÃO DIRIGIDA POR SINTAXE 4.1 Definições dirigidas por sintaxe 4.2 Atributos herdados e sintetizados 4.3 Esquemas de tradução dirigidos por sintaxe 4.4 Implementação	8
5	ANÁLISE SEMÂNTICA 5.1 O papel do analisador semântico 5.2 Verificação de tipo, classe e unicidade 5.3 Ações semânticas em derivadores LL e LR. 5.4 Grafo de dependência de atributos 5.5 Estrutura e organização de tabelas de símbolos 5.6 Aninhamento léxico, regras de escopo, descritores, verificação e conversão de tipos 5.7 Representação intermediária para análise semântica	8
6	GERAÇÃO DE CÓDIGO 6.1 Geração de código intermediário 6.2 Código de três endereços 6.3 Tipos e declarações 6.4 Expressões 6.5 Comandos 6.6 Procedimentos 6.7 Projeto de um gerador de código 6.8 Conjunto de instruções da linguagem objeto	8
7	GERAÇÃO DE CÓDIGO DE MÁQUINA 7.1 Seleção de instruções 7.2 Análise de tempo de vida 7.3 Alocação de registradores 7.4 Exemplos de otimização: otimização de laço	4
8	AMBIENTES EM TEMPO DE EXECUÇÃO 8.1 Organização de memória 8.2 Alocação de espaço da pilha 8.3 Gerenciamento de heap 8.4 Registros de ativação 8.5 Acesso a dados não locais 8.6 Passagem de parâmetro	4
Total		60

Bibliografia Básica

1	AHO, Alfred V. et al. Compiladores: princípios, técnicas e ferramentas. 2. ed. São Paulo: Pearson, 2007. E-book. (648 p.). ISBN 9788588639249. Disponível em: https://middleware-bv.am4.com.br/SSO/cefet/9788588639249 . Acesso em: 13 out. 2022.
2	COOPER, Keith D.; TORCZON, Linda. Construindo compiladores. 2. ed. Rio de Janeiro: LTC: Elsevier, c2014. 656 p. ISBN 9788535255645.
3	LOUDEN, Kenneth C. Compiladores: princípios e práticas. São Paulo: Cengage Learning, c2004. 569 p. ISBN 9788522104222; 8522104220.

Handwritten signature

Bibliografia Complementar	
1	CARTER, Nicholas. Teoria e problemas de arquitetura de computadores. Porto Alegre: Bookman, 2003. 240 p., il. (Coleção Shaum). ISBN 9788536302508 (broch.).
2	ORDONEZ, Edwards David Moreno; PENTEADO, Cesar Giacomini Penteado; SILVA, Alexandre César Rodrigues da Silva. Microcontroladores e FPGAs: aplicações em automação. São Paulo: Novatec, 2006.
3	PEREIRA, Fábio. Microcontroladores PIC: programação em C. 7. ed. São Paulo: Érica, 2007. 358 p. ISBN 9788571949355.
4	SIPSER, Michael. Introdução à teoria da computação. São Paulo: Cengage Learning, 2007. ISBN 9788522104994.
5	VIEIRA, Newton José. Introdução aos fundamentos da computação: linguagens e máquinas. São Paulo: Cengage Learning, c2006. 319 p. ISBN 9788522105083.


Prof.ª Gabriella Castro Barbosa Costa Dalpra
Coordenadora do Curso de Engenharia de Computação