



Universidade Federal de Goiás  
Escola de Engenharia Elétrica e da Computação  
Engenharia de Computação  
Matriz Curricular: ECOMP-ENG-2014

Plano de Disciplina

Ano 2019 - 1º Semestre

**Dados da Disciplina**

Código Disc.	Nome	Carga Horária	
		Teórica	Prática
136	COMPILADORES I	48	16

Prof Deborah Silva Alves Fernandes

Turma A

**Ementa**

Introdução à compilação. Fases da compilação. Ambiguidade. Relações sobre gramáticas. Análise sintática ascendente e descendente. Análise léxica. Tabelas de símbolos. Análise semântica e geração de código para uma máquina virtual. Introdução à otimização de código.

**Objetivo Geral**

Capacitar os alunos a compreender as fases do processo de compilação: análise léxica análise sintática, análise semântica. Propiciar aos alunos a oportunidade de desenvolver um compilador simplificado que permita o experimentar os aspectos principais envolvidos em cada fase de implementação de um compilador.

**Objetivos Específicos**

Estudar as técnicas envolvidas na análise léxica de uma linguagem de programação: modelagem de análise léxica por autômatos finitos, especificação de tokens por expressões regulares, obtenção automática de analisadores léxicos a partir de expressões regulares.  
Estudar gramáticas livres de contextos apropriadas para a análise sintática.  
Estudar o processo de geração automática de um analisador sintático a partir de uma gramática livre de contexto.  
Estudar análise semântica dirigida por sintaxe através dos formalismos de tradução dirigida por sintaxe e esquemas de tradução.  
Estudar os tipos de tradução dirigidas por sintaxes apropriados para ao tipo de análise sintática utilizado.  
Estudar implementações de tabelas de símbolos, geração de árvores sintáticas abstratas.  
Implementar os analisadores léxico, sintático e semântico e a geração de código para uma linguagem de programação.

**Relação com Outras Disciplinas**

O aluno deve aplicar os conhecimentos adquiridos nas disciplinas: Linguagens de Programação, Linguagens Formais e Autômatos, Arquitetura de Computadores (linguagens de montagem), Programação de Computadores I, Programação de Computadores II, Estruturas de Dados I, Estruturas de Dados II, e os conhecimentos adquiridos nesta disciplina para implementar um compilador com propósitos didáticos.

**Programa**

Introdução à compilação.



Fases da compilação.  
 Ambigüidade.  
 Relações sobre gramáticas.  
 Análise léxica.  
 Tabelas de símbolos.  
 Análise sintática ascendente e descendente.  
 Análise semântica e geração de código para uma máquina virtual.

### Procedimentos Didáticos

Legend	Descrição	Objetivo
AEX	Aula teórica	Transmitir conhecimento utilizando quadro ou slides.
RE	Aula teórica com resolução de exercícios	Desenvolver o raciocínio lógico, criatividade e capacidade de abstração e a capacidade de identificar, analisar e projetar soluções
TG	Trabalho em grupo	Desenvolver a capacidade de comunicação oral e escrita. Capacidade de trabalhar em grupo.
AP	Aula prática	Proporcionar ao aluno a aplicação prática do conteúdo ministrado em aula teórica.
ED	Estudo dirigido	Desenvolver a capacidade analítica, capacidade de síntese, de avaliação crítica e de análise.
SE	Seminários	Desenvolver o raciocínio lógico, criatividade, capacidade de abstração, capacidade para identificar, analisar, projetar soluções de problemas, a capacidade de comunicação oral e a capacidade de trabalhar em grupo.
OTR	Outros	Transmitir conhecimento utilizando quadro ou slides.

### Conteúdo Programático / Cronograma

Início	Proc. Didático	Tópico	# Aul.
14/03/19	AEX, OTR,	Apresentação da disciplina, notas e bibliografia. Visão geral da disciplina. Atividade Complementar: Inscrição na disciplina no sistema moodle e visualização do plano de ensino.	2
19/03/19	AEX, OTR,	Introdução - Tipos de tradutores, arquitetura de um compilador. Atividade Supervisionada: Leitura de material e entrega de resenha sobre conteúdo da aula (entregar trabalhos individuais e escritos à mão).	4
26/03/19	AEX, RE, AP, TG,	Análise léxica. Desenvolvimento de reconhecedores de expressões regulares com geradores de analisadores léxicos. (laboratório). Descrição do trabalho T1. Acompanhamento do desenvolvimento do trabalho T1. Tabela de símbolos. Resolução de exercícios. Atividade supervisionada: Lista de exercícios e leitura de material, desenvolvimento e entrega de resenha sobre conteúdo da aula (entregar trabalhos individuais e escritos à mão).	6
04/04/19	AEX, RE, ED, AP, TG	Análise sintática (gramáticas, analisadores bottom-up, conjuntos primeiro e seguinte, autômato de itens canônicos). Descrição do trabalho T2. Acompanhamento do desenvolvimento do trabalho T1. Atividades Complementares: 2 Listas de exercícios para entregar.	8
18/04/19	TG, AP, ED	Apresentação do trabalho T1.	4
25/04/19	AEX, RE, ED, AP, TG	Análise sintática - tabela sintática e algoritmo de análise SLR, tratamento de erros, implementação e exercícios. Analisadores LALR e LR canônico. Atividades Complementares: 2 Listas de exercícios para entregar.	4
02/05/19	OTR	Atividade avaliativa teórica 1.	2

Início	Proc. Didático	Tópico	# Aul.
09/05/19	AEX, RE, ED, AP, TG	Análise sintática top-down e aplicação de exercícios. Tratamento de erros na análise sintática. Acompanhamento do desenvolvimento do trabalho T2. Atividades Supervisionadas: Lista de exercícios sobre análise sintática top down para entregar. Leitura de material e entrega de resenha sobre tratamento de erros no analisador sintático.	4
16/05/19	AEX, RE, ED, AP, TG	Análise semântica, descrição do trabalho T3.	4
23/05/19	AEX, RE, ED, AP, TG	Apresentação do trabalho T2.	4
30/05/19	AEX, RE, ED, AP, TG	Tradução dirigida por sintaxe, acompanhamento trabalho T3. Atividades supervisionadas: Lista de exercícios de análise semântica para entregar.	8
13/06/19	AEX, RE, ED, AP, TG	Geração de código intermediário, otimização de código e geração de código. Resolução de Exercícios. Acompanhamento do desenvolvimento do trabalho T3. Atividades supervisionadas: Lista de exercício para entregar. Leitura de material e entrega de resenha sobre conteúdo visto em sala.	8
02/07/19	OTR	Atividade avaliativa teórica 2.	2
04/07/19	TG, AP, ED	Apresentação do trabalho T3.	4
Total			64

### Critério de Avaliação

As avaliações serão compostas por duas provas (P1 e P2), 8 atividades complementares (A1, ..., A8) e um trabalho de implementação de um pequeno compilador (TI). A nota final (NF) é obtida pelo cálculo:

$$NF = MP * 0,6 + TI * 0,4,$$

onde MP é dada por:

$$MP = (P1+AC1) + (P2+AC2) / 2$$

P1 e P2 : Provas escritas com valor máximo de 9,0 pontos

AC1 = Conjunto de atividades complementares 1 = A1+A2+A3+A4 = 1,0

AC2 = Conjunto de atividades complementares 2 = A5+A6+A7+A8 = 1,0

O (TI) trabalho de implementação é dividido em três etapas, conforme definição abaixo, e o cálculo da nota final do trabalho é dado por:

$$TI = (T1*0,25) + (T2*0,35) + (T3*0,4)$$

T1 – nota de apresentação do analisador léxico,

T2 – nota de apresentação do analisador sintático.

T3 – nota de apresentação do trabalho completo, incluindo a tradução dirigida por sintaxe e geração de código.

As notas dos trabalhos podem ter descontos nas entregas. Os descontos são de 0,3 pontos por dia de atraso.

Serão aprovados os alunos que obtiverem NF maior ou igual a 6,0 e tiverem frequência maior ou igual a 75% das horas-aulas do semestre.

Quanto ao requerimento de segunda chamada:

- Deverá ser protocolado na secretaria do INF ou no sistema SIGERA no prazo máximo de 5 dias úteis após à realização da prova, apresentando a comprovação da impossibilidade de seu comparecimento à primeira chamada da prova conforme condições estipuladas na Resolução CEPEC n. 1122/2012.

### Data da Realização das Provas

DATAS DAS PROVAS:

PROVA 1: 02/05/19

PROVA 2: 02/07/19

**ENTREGA DOS TRABALHOS:**

T1: 18/04

T2: 23/05

T3: 04/07

**Local de Divulgação dos Resultados das Avaliações**

Sala de aula e/ou área da disciplina na Plataforma de Educação a Distância do Instituto de Informática (Plataforma EaD-INF/UFG). A Plataforma EaD-INF/UFG é acessível no endereço eletrônico <http://ead.inf.ufg.br>.

**Bibliografia Básica**

- AHO, A.V.; ULLMAN, J.D. Compiladores – Princípios, Técnicas e Ferramentas, 2º Ed, 2007
- PRICE, A. M. A. - Implementação de linguagens de programação. 3º Ed. 2005
- DELAMARO, M. - Como construir um compilador utilizando ferramentas Java. Novatec. 2004

**Bibliografia Complementar**

- LOUDEN, K. C. Compiladores : princípios e práticas, 1º Ed, 2004
- FRASER, C. W. - A retargetable C compiler : design and implementation - 1º Ed . Addison-Wesley Professional. 1995
- MUCKNICK, S.S - Advanced compiler design and implementation - Ed. Morgan Kaufman, 1997
- RONALD, M. - Writing compilers and interpreters. 2º Ed. Wiley Computer Publishing. 1996
- HOLMES, J. - Building your own compiler with C++. Prentice Hall. 1995.

**Bibliografia Sugerida**

- AHO, Alfred V. and LAM, Monica S. , Compilers: principles, techniques and tools, 2nd. ed., Addison Wesley, 2006.
- COOPER, K. and TORCZON, L., Engineering a compiler, 2nd. ed., Morgan Kaufmann, 2011.
- GRUNE, D. et al., Modern compiler design, 2nd. ed., Springer, 2012.
- PATTERSON, David A. e HENNESSY, John. L.,- Organização e projeto de computadores: a interface hardware/software (tradução da 4ª edição original), 4. ed., Elsevier, 2013.

<b>Termo de Entrega</b>	<b>Termo de Aprovação</b>
Apresentado à Coordenação no dia	Aprovado em Reunião de CD no dia
Prof(a) Deborah Silva Alves Fernandes <i>Professor</i>	<i>Prof. Dr. Sérgio Teixeira de Carvalho</i> <i>Diretor do Instituto de Informática</i>
<b>Termo de Homologação</b>	
Data de Expedição: Goiânia, ____ de _____ de _____.	

