



PLANO DE ENSINO

1. IDENTIFICAÇÃO

Curso: Ciência da Computação - Diurno

Componente Curricular: Linguagens Formais e Autômatos

Fase: 5 (cinco)

Ano/Semestre: 2014/1

Numero de Créditos: 4 (quatro)

Carga horária - Hora Aula: 72 horas

Carga horária - Hora Relógio: 60 horas

Professor: Braulio Adriano de Mello

Atendimento ao Aluno: Segunda e quinta entre 18:10 e 19:10

2. OBJETIVO GERAL DO CURSO

O curso tem por objetivo a formação integral de novos cientistas e profissionais da computação, os quais deverão possuir conhecimentos técnicos e científicos e serem capazes de aplicar estes conhecimentos, de forma inovadora e transformadora, nas diferentes áreas de conhecimento da Computação. Adicionalmente, os egressos do curso deverão ser capazes de adaptar-se às constantes mudanças tecnológicas e sociais, e ter uma formação ao mesmo tempo cidadã, interdisciplinar e profissional.

3. EMENTA

Alfabeto, linguagens, gramáticas e suas representações. Gramáticas: definição formal, classificação (Hierarquia de Chomsky), propriedades. Gramáticas regulares, autômatos finitos, conjuntos regulares e expressões regulares. Gramáticas livres de contexto. Autômatos de pilha. Linguagens sensíveis ao contexto e irrestritas. Análise léxica e sintática.

4. OBJETIVOS

4.1. GERAL

Compreender e aplicar os fundamentos de teoria de linguagens na forma de máquinas geradoras e reconhecedoras.

4.2. ESPECÍFICOS

Domínio dos fundamentos de hierarquia de linguagens;
Compreensão dos teoremas para construção de máquinas formais;
Identificação e uso de formalismos e teoremas conforme classificação de linguagens;
Compreensão e aplicação dos métodos de análise de linguagens.

5. CRONOGRAMA E CONTEÚDOS PROGRAMÁTICOS

ENCONTRO	CONTEÚDO
1	Apresentação da disciplina, plano de aula, sistema de avaliação
2	Conceitos iniciais: Autômatos, Computabilidade (Church), decidibilidade (paradoxo de Russel), noções de compiladores
3	Teoria de linguagens: Símbolo, alfabeto, cadeia, sentença, concatenação, fechamento, sufixo, prefixo
4	Teoria de linguagens: Símbolo, alfabeto, cadeia, sentença, concatenação, fechamento, sufixo, prefixo
5	Definição formal de gramática gerativa e derivação Tipos de gramáticas, construção de gramáticas
6	Definição formal de gramática gerativa e derivação Tipos de gramáticas, construção de gramáticas
7	Construção de gramáticas e regras de derivação
8	Linguagens definidas por gramáticas, hierarquia de linguagens (Chomsky), máquinas reconhecedoras
9	Construção de gramáticas
10	Construção de gramáticas (exercícios)
11	Construção de gramáticas (exercícios)
12	Gramáticas Regulares / diagrama de transição / tabela de transição
13	Gramáticas Livres de Contexto
14	Gramáticas Livres de Contexto (exercícios)
15	Gramáticas Sensíveis ao Contexto e Irrestritas (Máquina de Turing como máquina reconhecedora de linguagens definidas por GSC e gramáticas irrestritas). Linguagens Enumeráveis Recursivamente e Recursivas.
16	Linguagens regulares, expressões regulares
17	Construção de expressões regulares
18	Autômato Finito (AFD e AFND), Teorema de determinização (Trabalho 1 : implementar o teorema de determinização. Entrada: Gramática regular não determinística. Saída: Autômato finito determinístico)
19	Eliminação de épsilon transições em AFDs
20	Revisão, exercícios
21	Avaliação 1
22	Minimização de AFDs
23	Processo para a construção de analisadores léxicos
24	Laboratório – impl. teorema de determinização
25	Apresentação do Trabalho 1
26	Linguagens Livres de Contexto e Árvores de derivação (analisadores sintáticos)
27	Máquina reconhecedora (Autômato de Pilha), reconhecedores

	ascendentes e descendentes
28	Teoremas de simplificação de GLCs: Eliminação de símbolos improdutivos.
29	Teoremas de simplificação de GLCs: Eliminação épsilon produções
30	Teoremas de simplificação de GLCs: Construção dos conjuntos First e Follow. (Trabalho 2 : Implementar o teorema. Entrada: GLC. Saída: conjunto first e conjunto follow da GLC)
31	Teoremas de simplificação de GLCs: Eliminação de produções unitárias
32	Teoremas de simplificação de GLCs: Eliminação de recursão a esquerda e
33	Fatoração de GLC.
34	Laboratório – implementação trabalho 2
35	Revisão e Avaliação 2
36	Apresentação do Trabalho 2
	Prova substitutiva – recuperação de rendimento (todo o conteúdo visto na disciplina).

6. PROCEDIMENTOS METODOLÓGICOS

Conduzir a disciplina com aulas expositivas/dialogadas enquanto discutidos os itens de cunho teórico, evoluindo em tópicos específicos para exercícios práticos, demonstrações, contextualização. Atividades práticas em implementação e experimentação.

7. AVALIAÇÃO DO PROCESSO ENSINO-APRENDIZAGEM

Uso de abordagens tais como: provas teóricas, avaliação escrita em aula, exercícios extra-classe, trabalhos de implementação, elaboração de texto/artigo, seminários entre outros trabalhos com complexidade variada.

Avaliação de trabalhos: 50% da nota pela parte escrita (grupo) e 50% da nota em avaliação individual (apresentação).

Avaliações da disciplina:

NP1: uma prova teórica (0,8) e um trabalho de implementação (0,2)

NP2: uma prova teórica (0,8) e um trabalho de implementação (0,2)

7.1 RECUPERAÇÃO: NOVAS OPORTUNIDADES DE APRENDIZAGEM E AVALIAÇÃO

Ao final do semestre, caso não tenha sido alcançada pontuação suficiente para aprovação, o estudante poderá usufruir de duas oportunidades de recuperação:

- Avaliação escrita adicional (todo o conteúdo do semestre) cuja nota será utilizada para substituir a menor nota dentre as avaliações regulares.
- Trabalhos facultativos individuais que, se entregues até o último dia letivo de aula da disciplina, podem ser utilizados para aumento da média final limitado a 3 pontos num total de 100. Trabalhos não realizados não implicam em redução de nota.

8. REFERÊNCIAS

8.1 BÁSICA

HOPCROFT, J. F., ULLMAN, J. D., Motwani, R., “Introdução à teoria dos autômatos”, Ed. Campus, 2002.
MENESES, P. B., “Linguagens Formais e Autômatos”, Ed. Sagra Luzzato, 2. edição, 1998.
AHO, A. V., SETHI, R., LAM, M., “Compiladores: Princípios, técnicas e ferramentas”, Ed. Longman do Brasil, 2007.

8.2 COMPLEMENTAR

PRICE, Ana M. Alencar e Toscani, Simão Sirineo. Implementação de Linguagens de Programação: Compiladores. Bookman Companhia Ed., 2008.
WOOD, D. , “Theory of Computation”, Ed. John Wiley & Sons, 1987.
SUDKAMP, T. A., “Languages and Machines – An Introduction to the Theory of Computer Science, 2. edição”, Ed. Addison Wesley, 1997.
LEWIS, H. R. e PAPADIMITRIOU, C. H. , “Elementos de Teoria da Computação”, Ed. Bookmam, 2. edição, 1998.
SIPSER, Michael. Introdução à Teoria da Computação. 2a ed., Thomson, 2007.
DIVERIO, T. A., MENEZES, P. B., “Teoria da Computação – Máquinas Universais e Computabilidade”, Ed. Sagra Luzzatto, Porto Alegre, 1999.

8.3 SUGESTÕES

Professor

Coordenador do curso