



Plano de Ensino

1 Código e nome da disciplina

WYF1513 TEORIA DA COMPUTAÇÃO E COMPILADORES

2 Carga horária semestral

3 Carga horária semanal

4 Perfil docente

5 Ementa

Nesta disciplina o aluno desenvolverá a habilidade de utilizar os conceitos fundamentais da teoria da computação, desde a teoria de linguagens formais até diferentes tipos de autômatos, para a análise e resolução eficaz e eficiente de problemas computacionais. Também poderá descrever as diferentes partes de um compilador e implementar analisadores léxicos e sintáticos para linguagens de programação. Além disso, o aluno utilizará a máquina de Turing para explorar os conceitos básicos de computabilidade.

6 Objetivos

1. Resolver de forma eficaz e eficiente problemas computacionais, utilizando os fundamentos e formalismos da teoria de computação. 2. Definir e analisar linguagens de programação, aplicando os conceitos da teoria de linguagens formais. 3. Identificar os conceitos e métodos mais importantes para a construção das fases de análise e síntese de um compilador. 4. Implementar um analisador léxico e um analisador sintático para uma linguagem de programação simples. 5. Utilizar a máquina de Turing e suas extensões para explorar os princípios básicos de computabilidade.

7 Procedimentos de ensino-aprendizagem

8 Temas de aprendizagem

1. Noções e terminologias matemáticas: representação e operações de conjuntos; funções e relações;

grafos e árvores; técnicas básicas de demonstração de provas matemáticas: prova por indução, prova por contradição. 10. Autômatos de pilha: definição de autômato de pilha determinístico e não-determinístico; processo de execução do autômato de pilha; exemplos; equivalência entre autômatos de pilha e gramáticas livres de contexto. 11. Análise sintática redutiva (ascendente): vantagens e desvantagens; operações e funcionamento; analisadores LR(k); tipos de analisadores LR(k); algoritmo para construir tabela de análise para analisador SLR(1). 12. Propriedades das linguagens livre de contexto: propriedade de fechamento; pertença, equivalência e finitude das linguagens livres de contexto; identificação de linguagens não livres de contexto usando o lema de bombeamento. 13. Máquina de Turing: definição e funcionamento da máquina de Turing; máquina de Turing como reconhecedores; máquina de Turing como transdutores; combinando máquina de Turing em tarefas complicadas; variantes e extensões da máquina de Turing. 14. Computabilidade e a máquina de Turing: a tese de Church-Turing; exemplos de problemas insolúveis em máquina de Turing; o problema de parada em máquina de Turing; aceitabilidade, decidibilidade e reducibilidade. 2. Introdução à teoria de linguagens formais: conceitos básicos: alfabeto, sentenças, sentença vazia, potência de sentença, fechamento de alfabeto; definição de linguagens formais; definição formal de gramática; derivações. 3. Linguagens de programação: evolução; padrões internacionais; tipos de tradutores; estrutura de um compilador: fase de análise e fase de síntese, tabela de símbolos e atendimento de erros; ferramentas para construção de compiladores. 4. Autômatos finitos (AF): autômato finito determinístico (AFD); autômato finito não-determinístico (AFND); diagrama de transição; tabela de transições; equivalência entre AFD e AFND; minimização de estados de AF. 5. Linguagens, expressões e gramáticas regulares: definição de linguagem regular; definição de expressão regular; operações regulares; definição de gramática regular; equivalência entre gramática regular, autômato finito e expressão regular. 6. Propriedades de linguagens regulares: propriedade de fechamento das linguagens regulares; pertença, equivalência e finitude das linguagens regulares; identificação de linguagens não regulares usando o lema de bombeamento. 7. Análise léxica: definição de símbolos léxicos válidos de uma linguagem de programação usando gramática regulares, expressões regulares ou autômatos finitos; construção de tabela de símbolos; implementação de um analisador léxico. 8. Gramáticas livres de contexto (GLC): representação e manipulação; ambiguidade; associatividade e precedência de operadores; GLC sem ciclos -livre, fatorada à esquerda, recursiva à esquerda e simplificada; algoritmos para transformação de GLC. 9. Análise sintática descendente: requisitos de gramática para construir analisadores descendentes; analisador recursivo preditivo; analisador tabular preditivo; implementação de analisador recursivo preditivo.

9 Procedimentos de avaliação

10 Bibliografia básica

AHO, A. V.; SETHI, R. ULLMAN, J. J. Compiladores: Princípios, técnicas e ferramentas. Rio de Janeiro: LTC, 1995. DIVERIO, Tiaraju A.; MENEZES, Paulo F. Blauth. Teoria da Computação – Máquinas Universais e Computabilidade. Porto Alegre: Sagra-Luzzatto, 2000. SIPSER, Michael. Introdução À Teoria da Computação. Thomson Pioneira.

11 Bibliografia complementar

GRUNE, D. et al. Projeto moderno de compiladores: Implementação e aplicações. Rio de Janeiro: Campus, 2001. LEWIS, Harry R. Elementos de Teoria da Computação. Porto Alegre: Makron Books, 2004. HOPCROFT, John E. Introdução a Teoria de Autômatos, Linguagens e Computação. Rio de Janeiro: Record, 2002. LOUDEN, Kenneth C. Compiladores - Princípios e Práticas. Thomson. PRICE, Ana Maria Alencar. Implementação de linguagens de programação: compiladores. Porto Alegre: Editora Sagra Luzatto, 2005.

