



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO



FICHA DE DISCIPLINA

DISCIPLINA: ABSTRAÇÃO DE DADOS E PROVA AUTOMÁTICA DE PROGRAMAS

CÓDIGO: GBC200		UNIDADE ACADÊMICA: FACULDADE DE COMPUTAÇÃO		
PERÍODO/SÉRIE: -		CH TOTAL TEÓRICA:	CH TOTAL PRÁTICA:	CH TOTAL:
OBRIGATORIA: ()	OPTATIVA: (X)	60	00	60
NÚCLEO DE FORMAÇÃO: Tecnológica / Profissional				
PRÉ-REQUISITOS: NÃO HÁ		CÓ-REQUISITOS: NÃO HÁ		

OBJETIVOS

O objetivo geral do curso é o estudo e utilização de técnicas de prova automática aplicadas à verificação de sistemas e programas. São tratadas três técnicas de verificação: Síntese e Análise de Semântica de Programas, Análise Estática e Análise Dinâmica. Assim, ao final do curso o aluno deverá estar habilitado nas principais técnicas de verificação programas, incluindo correção por construção, prova de programas, análise estática e dinâmica.

EMENTA

Conceitos gerais sobre Especificação, Verificação e Teste de Programas. Provas de Programas utilizando o paradigma VDM. Análise Estática. Análise Dinâmica.

DESCRIÇÃO DO PROGRAMA

1. **Introdução**
 - 1.1. *O que é especificação de programa* ou “o que o programa deve fazer”.
 - 1.2. *O que é verificação de programa* ou “como se certificar de que o programa vai fazer corretamente o que foi especificado para fazer”.
 - 1.3. *O que é teste de programa* ou “como mostrar que o programa contém erros”.
 - 1.4. O que é *debugging* ou “como encontrar as causas dos erros”
 - 1.5. O que é análise estática ou “o que se pode dizer sobre o programa sem executá-lo”
 - 1.6. Os métodos SAT
2. **Análise Semântica – Conceitos Gerais**
 - 2.1. Pré-condição e Pós-condição
 - 2.2. Princípio de Refinamento Top-Down
3. **Prova de Correção de Programas sem Laços (paradigma VDM)**

- 3.1. Correção de Programas – Definição
- 3.2. Pré-condição mais fraca (wp(S,Q))
- 3.3. Experimentos SPARK
- 3.4. Derivação da Pré-condição parcial mais fraca
- 3.5. Método Assertivo
- 4. Prova de Programas Iterativos (paradigma VDM)**
 - 4.1. Programas com Laços
 - 4.2. Invariantes de Laços
 - 4.3. Invariantes de Objetivo
- 5. Testes de Caixa-Preta**
 - 5.1. Testes de Princípios
 - 5.2. Testes de Funcionalidade
 - 5.3. Testes Aleatórios
- 6. Análise Estática**
 - 6.1. Parse de Programas e Árvores Sintáticas
 - 6.2. Flowgraph de controle
 - 6.3. Derivação de Flowgraph
 - 6.4. Caminhos no Flowgraph
 - 6.5. Dependências
 - 6.6. Controle de Dependências: controle estruturado e arbitrário
 - 6.7. Anomalias de Fluxos de Controle
 - 6.8. Anomalias de Fluxos de Dados
 - 6.9. Análise Estática Descritiva
- 7. Análise Dinâmica**
 - 7.1. Conceitos de Análise Dinâmica
 - 7.2. Teste Estrutural de Programas
 - 7.3. Dependências de Programas em Testes de Software
 - 7.4. Semântica Operacional
 - 7.5. Aplicações

BIBLIOGRAFIA

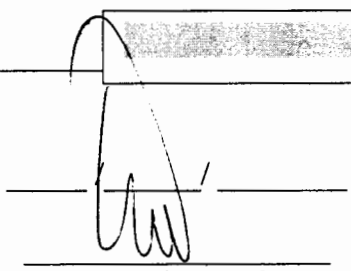
Básica

LASKI, J., STANLEY, J. : Software Verification and Analysis: An Integrated, Hands-On Approach. Springer Verlag, 1ª Edição, 2009.
NIELSON, F., NIELSON, H.R., HANKIN, C.: Principles of Program Analysis. Springer Verlag, 2004.

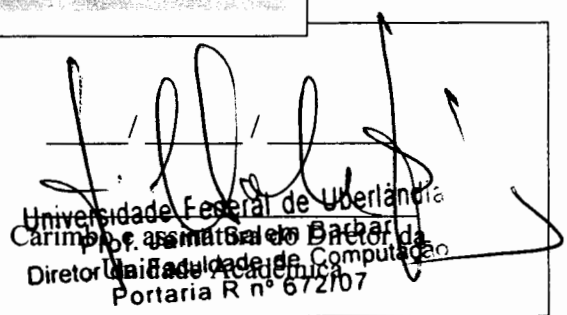
Complementar

HUTH, R. A; RYAN, M. D. Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, 2ª Edição 2004.
JACKSON, D. : Software Abstractions: Logic, Language and Analysis. The MIT Press, Cambridge, Massachusetts. 2006.

APROVAÇÃO



Carimbo e assinatura do Coordenador do curso
Universidade Federal de Uberlândia
Prof. Ilmério Reis da Silva
Coordenador do Curso de Ciência da Computação
Portaria R nº 711/09



Universidade Federal de Uberlândia
Carimbo e assinatura do Diretor da
Unidade Acadêmica
Portaria R nº 672/07