

# Lógica da Computação

## Aula 01

Raul Ikeda

2º semestre de 2018

# Objetivos de Aprendizagem

1. Especificar uma gramática para reconhecer uma linguagem de interesse.
2. Saber programar e gerar o diagrama explicativo de um autômato que implementa uma gramática.
3. Especificar um analisador léxico por meio de expressões regulares.
4. Compreender os conceitos básicos sobre Lógica Matemática, Teoria da Computabilidade e Máquina de Turing.
5. Entender todos os passos necessários à compilação de programas.

# Mapeamento das Avaliações

- LING - LINGUAGEM:
  - ✓ Especificar uma gramática para reconhecer uma linguagem de interesse.
- COMP - COMPILADOR:
  - ✓ Saber programar e gerar o diagrama explicativo de um autômato que implementa uma gramática.
  - ✓ Especificar um analisador léxico por meio de expressões regulares.
  - ✓ Entender todos os passos necessários à compilação de programas.
- PROV - PROVAS:
  - ✓ Saber programar e gerar o diagrama explicativo de um autômato que implementa uma gramática.
  - ✓ Compreender os conceitos básicos sobre Lógica Matemática, Teoria da Computabilidade e Máquina de Turing.

# Avaliações: LING - Linguagem

- Durante o curso será desenvolvida uma linguagem de programação com características proprietárias.
- Projeto em caráter de APS e individual.
- Requisitos:
  1. Adequar a linguagem à uma GLC e estruturá-la segundo o padrão EBNF.
  2. Utilizar as ferramentas Flex e Bison (ou semelhantes) para realizar as etapas de Análise Léxica e Sintática.
  3. Utilizar a LLVM (ou derivadas) para implementar a sua linguagem até a fase final de compilação. Não é preciso implementar um compilador novo.
  4. Criar um exemplo de testes que demonstre as características da sua Linguagem.
  5. Fazer uma apresentação de 15 minutos na data da entrega final.
- Nota da avaliação:
  - A+: se cumprir todos os requisitos acima.
  - C: se cumprir os requisitos, exceto o número 3.
  - D: se não cumprir os requisitos 1, 2, 4 ou 5.
  - I: se não houver entrega ou se for irrelevante.

# Avaliações: COMP - Compilador

- Durante o semestre será desenvolvido incrementalmente um Compilador da linguagem C, aplicando o estudo de compiladores. No total serão 10 atividades com entregas.
- Requisitos:
  - Individual. Uso do Git obrigatório.
  - Desenvolvimento em aula estúdio. As tarefas pendentes terão prazos de entrega entre 1 e 2 semanas dependendo do roteiro.
  - Usar uma linguagem orientada a objetos com recursividade e dicionários.
- Nota:
  - I se não entregar nenhuma atividade.
  - D se não entregar pelo menos uma atividade.
  - A se não atrasar nenhuma entrega.
  - Para cada atraso na entrega, desconta-se  $\frac{1}{4}$  de conceito, arredondando para baixo no final do semestre.
  - 11ª Entrega extra: adiciona meio conceito.
  - Será permitido o atraso de até 1 relatório sem ônus na nota.

# Avaliações: PROV - Prova

- 40% - AI:
  - Teoria de Conjuntos.
  - Linguagens Regulares.
  - Linguagens Livres de Contexto.
  - Autômatos Finitos e de Pilha.
  - Hierarquia de Chomsky.
- 60% - AF:
  - Computabilidade.
  - Máquina de Turing.
  - Decidibilidade e Intratabilidade.
  - Linguagens Sensíveis ao Contexto e Recursivamente Enumeráveis.
  - Complexidade.
  - Lógica Proposicional e de Predicados.
  - Prova Matemática e verificação de programas.

# Nota Final

- Se  $LING$ ,  $COMP$  e  $PROV \geq 5$  ou C:

$$NF = 0,2LING + 0,4COMP + 0,4PROV$$

- Caso contrário:

$$NF = \min(LING, COMP, PROV)$$

Será usada a tabela oficial de conversão de conceito para nota do Blackboard.

# Bibliografia

- Compiladores:
  - JOSÉ NETO, J., Introdução à Compilação., 1ª ed., Elsevier, 2016
  - AHO, A.V.; LAM, M. S.; SETHI, R.; ULLMAN, J., Compiladores: Princípios, Técnicas e Ferramentas, 2ª ed., Longman, 2007
- Linguagens Formais e Autômatos:
  - RAMOS, M. V. M.; JOSÉ NETO, J.; VEJA, I. S., Linguagens Formais. Teoria, Modelagem e Implementação, 1ª ed., Bookman, 2009
  - HOPCROFT, J. E.; ULLMAN, J. D; MOTWANU, R., Introdução à Teoria dos Autômatos, Linguagens e Computação, 1ª ed., CAMPUS, 2002
- Computabilidade e Lógica:
  - BOOLOS, G. S.; BURGESS, J. P.; JEFFREY, R. C., Computabilidade e Lógica, 1ª ed., Unesp, 2012
  - SILVA, F. C.; FINGER, M.; MELO, A. C. V., Lógica para Computação, 2ª ed., Cengage, 2017
- Complexidade:
  - CORMEN, Thomas H., LEISERSON; Charles E., RIVEST; Ronald L., STEIN, Clifford., Algoritmos: teórica e prática., 3ª ed., Elsevier-Campus., 2012
  - SIPSER, M., Introdução à Teoria da Computação, 2ª ed., Thomson Pioneira, 2007



# Próxima Aula

- Discussão do artigo: Neto, J. J. A Teoria da Computação e o profissional de informática. Revista de Computação e Tecnologia da PUC-SP, vol. 1, 2009.
  - Revista Open Access, mas publicado no Blackboard.
  - Duração: até 30 minutos no início.
- Teoria dos Conjuntos e Linguagens Formais
  - Ramos et al. Cap 1 e 2.
  - Neto, J. J. Cap 2

# Atividade I – 30 minutos

- Você foi incumbido de criar uma linguagem de programação de **ALTA PRODUTIVIDADE**.
- Criar uma wishlist das features desejadas, justificando a escolha.
- Tentar organizar a linguagem em algum esquema ou diagrama.
- Requisito mínimo:
  - A linguagem deve permitir implementar qualquer algoritmo de programação.
  - Deve conter no mínimo: variáveis, funções, condicionais, loops, I/O.
  - Desejável: classes (POO), vetores e características singulares (diretivas de compilação, funcional, etc)
- Entregar o material produzido durante a aula.

# Atividade II – 1 hora

- Roteiro Zero do Compilador.
- Lembrando:
  - Individual.
  - Usar uma linguagem orientada a objetos com recursividade e dicionários.