

# DESENVOLVIMENTO DE UM COMPILADOR BASEADO EM LINGUAGENS FORMAIS E AUTÔMATOS

Kauã Fernandez Freitas

Unic Beira Rio

Palavras-chave: compiladores; linguagens formais; autômatos; análise sintática; teoria da computação.

## 1. Introdução

### 1.1. Apresentação do Tema e Relevância

Os compiladores desempenham um papel fundamental na área da Ciência da Computação, atuando como tradutores entre linguagens de alto nível e linguagens de máquina. Eles utilizam princípios de linguagens formais e autômatos para realizar a análise léxica, sintática e semântica de programas, garantindo que o código-fonte seja corretamente interpretado e executado.

conceitos de linguagens formais e autômatos podem ser aplicados no desenvolvimento de um compilador funcional, capaz de realizar as etapas de análise léxica, sintática e semântica de um programa.

Embora existam compiladores amplamente difundidos e otimizados, muitos deles são tratados como “caixas pretas”, dificultando a compreensão do funcionamento interno dos processos de tradução de código. Essa lacuna teórica se torna ainda mais evidente em ambientes acadêmicos, onde o entendimento prático da relação entre teoria da computação e implementação de compiladores é essencial para a formação de desenvolvedores e engenheiros de software.

### 1.2. Definição do Problema de Pesquisa

O problema central deste trabalho consiste em compreender como os

Portanto, o problema de pesquisa que orienta este estudo é: como desenvolver um compilador baseado em conceitos de linguagens formais e autômatos que seja capaz de reconhecer, analisar e interpretar

**REFERÊNCIAS:** AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compiladores: Princípios, Técnicas e Ferramentas**. 2. ed. São Paulo: Pearson, 2008.

CHOMSKY, Noam. **Three Models for the Description of Language**. *IRE Transactions on Information Theory*, v. 2, n. 3, p. 113–124, 1956.

# DESENVOLVIMENTO DE UM COMPILADOR BASEADO EM LINGUAGENS FORMAIS E AUTÔMATOS

Kauã Fernandez Freitas

Unic Beira Rio

Palavras-chave: compiladores; linguagens formais; autômatos; análise sintática; teoria da computação.

códigos escritos em uma linguagem de programação simplificada?

A resolução desse problema busca integrar o conhecimento teórico — envolvendo gramáticas, autômatos e teoria da computação — com sua aplicação prática na construção de um compilador didático, promovendo um aprendizado mais significativo sobre o funcionamento interno das linguagens de programação.

## 2. Objetivos

O objetivo deste artigo é demonstrar a relação entre compiladores e linguagens formais, explorando como os autômatos e as gramáticas fornecem a base teórica para o processo de compilação. Busca-se compreender como essas

estruturas formais se aplicam à análise e tradução de código.

## 3. Metodologia

A metodologia adotada neste trabalho caracteriza-se como uma pesquisa aplicada de natureza experimental, cujo objetivo é implementar e analisar o funcionamento de um compilador desenvolvido com base em conceitos teóricos de linguagens formais e autômatos.

O desenvolvimento do compilador foi dividido em etapas, correspondentes às principais fases do processo de compilação:

**Análise Léxica:** implementação de um módulo responsável por identificar os tokens da linguagem — palavras reservadas, operadores, identificadores e literais — utilizando princípios de **autômatos finitos determinísticos (AFD)**.

**REFERÊNCIAS:** AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compiladores: Princípios, Técnicas e Ferramentas**. 2. ed. São Paulo: Pearson, 2008.

CHOMSKY, Noam. **Three Models for the Description of Language**. *IRE Transactions on Information Theory*, v. 2, n. 3, p. 113–124, 1956.

# DESENVOLVIMENTO DE UM COMPILADOR BASEADO EM LINGUAGENS FORMAIS E AUTÔMATOS

Kauã Fernandez Freitas

Unic Beira Rio

Palavras-chave: compiladores; linguagens formais; autômatos; análise sintática; teoria da computação.

**Análise Sintática:** construção de um **analisador sintático** baseado em **gramáticas livres de contexto (GLC)**, verificando se a sequência de tokens segue as regras sintáticas da linguagem.

**Análise Semântica:** criação de rotinas de verificação de tipos, escopo e coerência lógica, assegurando que o código-fonte possua significado válido dentro das regras da linguagem.

**Geração e Interpretação de Código:** em vez de gerar código de máquina, o compilador desenvolvido possui um **interpretador**, responsável por executar diretamente as instruções reconhecidas, permitindo observar o comportamento do programa em tempo real.

Os resultados obtidos com o desenvolvimento do compilador descrito demonstraram que os conceitos de linguagens formais e autômatos — empregados nas fases de análise léxica, sintática e semântica — funcionam efetivamente como base para a construção de uma ferramenta de tradução e execução de código-fonte.

Foram selecionados modelos de código-fonte simples escritos na linguagem proposta para validação do compilador. Para facilitar o processo, utilizou-se como referência o repositório The Super Tiny Compiler (JavaScript) que oferece uma implementação didática dos principais componentes de um compilador.

De modo geral, os resultados confirmaram que um compilador baseado em linguagens formais e autômatos pode ser

## 4. Resultado

**REFERÊNCIAS:** AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compiladores: Princípios, Técnicas e Ferramentas**. 2. ed. São Paulo: Pearson, 2008.

CHOMSKY, Noam. **Three Models for the Description of Language**. *IRE Transactions on Information Theory*, v. 2, n. 3, p. 113–124, 1956.

# DESENVOLVIMENTO DE UM COMPILADOR BASEADO EM LINGUAGENS FORMAIS E AUTÔMATOS

Kauã Fernandez Freitas

Unic Beira Rio

Palavras-chave: compiladores; linguagens formais; autômatos; análise sintática; teoria da computação.

implementado com sucesso em um contexto educacional, oferecendo não apenas uma ferramenta prática, mas também um suporte direto à aprendizagem teórica. O código disponibilizado comprova a viabilidade e serve como base para futuras extensões, como geração de código nativo, suporte a tipos avançados e otimizações.

## 5. Conclusão

O desenvolvimento de um compilador baseado em linguagens formais e autômatos permitiu compreender de forma prática como os conceitos teóricos da disciplina de Compiladores se interligam no processo de tradução de linguagens. A implementação de cada etapa — análise léxica, sintática e semântica — demonstrou que o funcionamento de um compilador segue princípios matemáticos sólidos,

fundamentados em autômatos finitos, gramáticas livres de contexto e linguagens formais.

A utilização do projeto *The Super Tiny Compiler* como referência experimental mostrou-se eficaz para validar a aplicabilidade dos conceitos teóricos estudados. Por meio dele, foi possível observar a construção e funcionamento de um compilador em sua essência, desde o reconhecimento de tokens até a geração de uma representação intermediária do código. Esse modelo evidenciou, de maneira clara e acessível, como um código-fonte pode ser decomposto e interpretado, reforçando a importância da formalização das linguagens no desenvolvimento de sistemas computacionais.

Conclui-se que o estudo e a prática do desenvolvimento de compiladores vão muito além da implementação de ferramentas; representam um exercício de

**REFERÊNCIAS:** AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compiladores: Princípios, Técnicas e Ferramentas**. 2. ed. São Paulo: Pearson, 2008.

CHOMSKY, Noam. **Three Models for the Description of Language**. *IRE Transactions on Information Theory*, v. 2, n. 3, p. 113–124, 1956.

# DESENVOLVIMENTO DE UM COMPILADOR BASEADO EM LINGUAGENS FORMAIS E AUTÔMATOS

Kauã Fernandez Freitas

Unic Beira Rio

Palavras-chave: compiladores; linguagens formais; autômatos; análise sintática; teoria da computação.

compreensão profunda sobre como a linguagem humana é traduzida em instruções compreensíveis por máquinas. O domínio dos fundamentos teóricos — linguagens formais, autômatos e gramáticas — é essencial para a formação de profissionais capazes de entender e projetar sistemas complexos, otimizando processos de tradução, execução e interpretação de programas.

Em síntese, o projeto demonstrou que a ponte entre teoria e prática é não apenas possível, mas necessária para o avanço do conhecimento em Computação, reforçando o papel das linguagens formais como base estrutural da ciência e da tecnologia da informação.

**REFERÊNCIAS:** AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compiladores: Princípios, Técnicas e Ferramentas**. 2. ed. São Paulo: Pearson, 2008.

CHOMSKY, Noam. **Three Models for the Description of Language**. *IRE Transactions on Information Theory*, v. 2, n. 3, p. 113–124, 1956.