

Gustavo P. Gouveia (6482819), Victor Lassance (6431325)

**Relatório de Compiladores**  
**Quinta Etapa**  
**Tradução dos Comandos**  
**Linguagem de programação CZAR**

Texto apresentado à Escola Politécnica da Universidade de São Paulo como requisito para a aprovação na disciplina Linguagens e Compiladores no quinto módulo acadêmico do curso de graduação em Engenharia de Computação, junto ao Departamento de Engenharia de Computação e Sistemas Digitais (PCS).

Universidade de São Paulo

Escola Politécnica

Engenharia de Computação - Curso Cooperativo

Professor: Ricardo Luis de Azevedo da Rocha

São Paulo

2013

# Sumário

	<b>Sumário</b> . . . . .	<b>2</b>
<b>1</b>	<b>Introdução</b> . . . . .	<b>3</b>
<b>2</b>	<b>Tradução de estruturas de controle de fluxo</b> . . . . .	<b>4</b>
<b>3</b>	<b>Tradução de comandos imperativos</b> . . . . .	<b>5</b>
<b>4</b>	<b>Cálculo de expressões aritméticas e booleanas</b> . . . . .	<b>6</b>
<b>5</b>	<b>Exemplo de programa traduzido</b> . . . . .	<b>7</b>
	<b>Referências</b> . . . . .	<b>8</b>

# 1 Introdução

Este projeto tem como objetivo a construção de um compilador de um só passo, dirigido por sintaxe, com analisador e reconhecedor sintático baseado em autômato de pilha estruturado.

Em um primeiro momento, foi definida uma linguagem de programação e identificados os tipos de átomos. Para cada átomo foi escrito uma gramática linear representativa da sua lei de formação e um reconhecedor para o átomo. Desse modo, as gramáticas assim escritas foram unidas e convertidas em um autômato finito, o qual foi transformado em um transdutor e implementado como sub-rotina, dando origem ao analisador léxico propriamente dito. Também foi criada uma função principal para chamar o analisador léxico e possibilitar o seu teste.

Durante a segunda etapa, a sintaxe da linguagem, denominada por nós de CZAR, foi definida formalmente a partir de uma definição informal e de exemplos de programas que criamos, misturando palavras-chave e conceitos de diferentes linguagens de programação. As três principais definições foram escritas na notação BNF<sup>1</sup>, Wirth<sup>2</sup> e com diagramas de sintaxe.

Na terceira etapa, implementamos o módulo referente à parte sintática para a nossa linguagem. O analisador sintático construído obtém uma cadeia de *tokens* proveniente do analisador léxico, e verifica se a mesma pode ser gerada pela gramática da linguagem e, com isso, constrói a árvore sintática (ALFRED; SETHI; JEFFREY, 1986).

Para a quarta entrega, focamos no ambiente de execução. O compilador por nós criado terá como linguagem de saída um programa que será executado na máquina virtual conhecida como Máquina de von Neumann (MVN).

Para a entrega atual, buscamos completar a especificação do código gerado pelo compilador e das rotinas do ambiente de execução da nossa linguagem de alto nível, a CZAR.

Como material de consulta, além de sites sobre o assunto e das aulas ministradas, foi utilizado o livro indicado pelo professor no começo das aulas (NETO, 1987), para pesquisa de conceitos e possíveis implementações.

O documento apresenta a seguir o que foi solicitado na quinta etapa.

---

<sup>1</sup> Ver [http://en.wikipedia.org/wiki/Backus\\_Naur\\_Form](http://en.wikipedia.org/wiki/Backus_Naur_Form)

<sup>2</sup> Ver [http://en.wikipedia.org/wiki/Wirth\\_syntax\\_notation](http://en.wikipedia.org/wiki/Wirth_syntax_notation)

## 2 Tradução de estruturas de controle de fluxo

Tradução de estruturas de controle de fluxo

- Desvio
- If-then (obrigatório)
- If-then-else (obrigatório)
- While (obrigatório)
- Do-until

Falar que o comando N pode ser substituído por  $+0$ , por exemplo, para simplificar o processo e tornar o código sequencial.

### 3 Tradução de comandos imperativos

Tradução de comandos imperativos

- Atribuição de valor (obrigatório)
- Leitura (entrada) (obrigatório)
- Impressão (saída) (obrigatório)
- Chamada de subrotina (obrigatório)

## 4 Cálculo de expressões aritméticas e booleanas

Cálculo de expressões - código alinhavado.

## 5 Exemplo de programa traduzido

Exemplo de programa traduzido

(apresentar exemplo de programa escrito em sua linguagem e sua respectiva tradução para a linguagem de saída)

# Referências

ALFRED, V.; SETHI, R.; JEFFREY, D. *Compilers: principles, techniques and tools*. [S.l.]: Addison-Wesley, 1986.

NETO, J. J. *Introdução à Compilação*. [S.l.]: LTC, 1987. (ENGENHARIA DE COMPUTAÇÃO).