



FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

Aspectos teóricos da computação

RELATÓRIO AV3

Lucas Duarte - 2110912

Gabriel Araújo - 2110916

Marcelo Feliciano Figueiredo - 2110871

Matheus Tavares 2110920

Davi Alencar - 2110884

Introdução

O projeto desenvolvido pelo grupo consiste em um transpilador que converte código escrito em Java para código Python. Um transpilador, também conhecido como fonte-para-fonte ou compilador de nível alto-para-nível alto, traduz o código fonte de uma linguagem de programação de alto nível para outra. Este projeto foi implementado usando JavaCC (Java Compiler Compiler), uma ferramenta que gera analisadores sintáticos em Java a partir de uma gramática formal.

Objetivo

O objetivo do transpilador é permitir que desenvolvedores escrevam código em Java e o converta automaticamente para Python, mantendo a lógica e a estrutura do código original. Isso é particularmente útil para equipes que precisam migrar código entre essas duas linguagens populares ou para aqueles que desejam aprender Python a partir de um código existente em Java.

Estrutura do Projeto

O projeto é dividido em várias partes principais, cada uma responsável por um aspecto específico da tradução:

1. **Gramática e Tokens:** Definição dos tokens e das regras gramaticais para o reconhecimento do código Java.
2. **Analisador Sintático:** Análise da estrutura do código Java para garantir que ele siga as regras da linguagem.
3. **Geração de Código:** Conversão das estruturas sintáticas reconhecidas para o equivalente em Python.
4. **Interface de Entrada e Saída:** Manipulação da entrada do código Java e a saída do código Python gerado.

Análise de Complexidade

A análise de complexidade do transpilador pode ser dividida em duas partes principais: a análise léxica e a análise sintática.

1. **Análise Léxica:**

- A complexidade da análise léxica é linear, ou seja, $O(n)$, onde n é o tamanho da entrada. O analisador léxico percorre a entrada uma vez, identificando tokens válidos.

2. **Análise Sintática:**

- A complexidade da análise sintática pode variar dependendo da gramática. No caso de gramáticas livres de contexto, como a usada neste transpilador, a complexidade pode ser $O(n^3)$ na pior das hipóteses com um analisador LL ou LR. No entanto, com otimizações e gramáticas bem definidas, pode-se alcançar uma complexidade média mais próxima de $O(n^2)$ ou mesmo $O(n)$.

Perguntas ao Chat

1. Estou com o erro: Warning: Choice conflict involving two expansions at line 122, column 7 and line 127, column 7 respectively. A common prefix is: Consider using a lookahead of 3 or more for earlier expansion. e Warning: Choice conflict involving two expansions at line 122, column 7 and line 127, column 7 respectively. A common prefix is: Consider using a lookahead of 3 or more for earlier expansion. me explique o que está acontecendo e como resolver
2. ta tendo um conflito entre 2 métodos variable declaration e methoddeclaration

Implementação

Link do GitHub

<https://github.com/MarceloFigDev/Traducao-de-Codigo-Java-para-Python>

CONCLUSÃO

O desenvolvimento de um transpilador de Java para Python representa uma solução prática e eficiente para a migração de código entre essas duas linguagens de programação amplamente utilizadas. O projeto foi estruturado utilizando JavaCC para criar um analisador sintático que reconhece a gramática Java e traduz suas estruturas para Python.

Os principais desafios enfrentados incluíram a definição precisa das regras gramaticais e a gestão da tradução correta das estruturas de controle, como loops, condicionais, declarações de variáveis, e inicialização de arrays. O projeto também precisou garantir que as expressões aritméticas e operações de concatenação de strings fossem traduzidas de forma adequada.

A análise de complexidade revelou que a abordagem adotada proporciona um desempenho eficiente para a maioria dos casos de uso, com a análise léxica operando em complexidade linear e a análise sintática podendo variar de $O(n^2)$ a $O(n^3)$ dependendo da complexidade do código de entrada.

O projeto ainda pode ser estendido para suportar mais características da linguagem Java e otimizações na tradução para Python. A capacidade de traduzir métodos e declarações de classes mais complexas pode ser adicionada para tornar o transpilador mais robusto e versátil.

A ferramenta desenvolvida tem grande potencial de aplicação em cenários educacionais, onde estudantes de programação podem aprender Python a partir de seu conhecimento prévio em Java. Além disso, equipes de desenvolvimento que necessitam migrar seus projetos entre essas duas linguagens podem se beneficiar significativamente dessa ferramenta, economizando tempo e esforço no processo de reescrita manual do código.

