

MiniJavaCompiler

Introdução

- Este projeto tem o objetivo de implementar um compilador para a linguagem Mini-Java, implementado em Java. Mini-Java é um subconjunto da linguagem Java, apresentada pelo livro *Modern Compiler Implementation in Java*. Assim, existem algumas restrições da linguagem em relação ao Java, como por exemplo, o comando `"System.out.println()"` só pode imprimir inteiros, outro exemplo é que o Mini-Java não aceita sobrecarga, entre outros. É possível achar mais informações sobre essa linguagem no site da linguagem (<http://www.cambridge.org/us/features/052182060X/index.html>). O projeto foi implementado ao longo da disciplina de Construção de Compiladores, ministrada pelo professor Heron de Carvalho, do Departamento de Computação da Universidade Federal do Ceará no período letivo 2010.1.

Desenvolvimento

- O desenvolvimento do trabalho deveria ocorrer no decorrer da disciplina, à medida que os capítulos do livro texto fossem sendo explanados pelo professor.

Módulo I: Análise Léxica e Parser

- Para o analisador léxico e *parser*, utilizamos a biblioteca indicada, *JavaCC*. Essa biblioteca gera arquivos em Java, a partir de um arquivo `".jj"`, onde declaramos os *tokens* da linguagem e escrevemos a gramática da linguagem gerando um *parser* preditivo. A gramática definida para gerar o *parser* foi baseada na gramática da linguagem apresentada pelo livro, porém, para evitar o uso de *lookahead* maior que 1, outros não-terminais foram adicionados com o objetivo de refatorar e remover recursões.

Módulo II: Análise Semântica

- Na linguagem, para cada produção de sua gramática, deve ser atribuído algum significado. Para isso, o *parser* preditivo gera uma árvore sintática abstrata, o uso dessa árvore traz o benefício de separar o Módulo I do Módulo II, fazendo com que a manutenção do compilador seja mais fácil e permitindo que a leitura de um programa passe inteiramente por uma análise sintática, para que depois, com sua árvore sintática gerada, a análise semântica seja realizada. Implementamos uma tabela de símbolos imperativa, onde informações sobre os tipos são mantidas. Conforme recomendado pelo livro, implementamos a checagem de tipos seguindo o padrão Visitor para separar a estrutura sintática das diferentes interpretações semânticas em diferentes Visitors.

Módulos Não-Implementados

- Os Módulos III (Representação Intermediária), IV (Construção de Blocos Básicos e Traços), V (Seleção de Instruções), VI (Análise de Longevidade), VII (Alocação de Registradores) e o módulo final, que consiste na junção de todos os módulos, não foram implementados.

Conclusão

- Foram encontradas algumas dificuldades durante a implementação do projeto. Como algumas dúvidas que surgiram sobre como a implementação deveria ser conduzida, apesar da disponibilidade do professor. Mas a parte implementada serviu para fixar ainda mais os conhecimentos adquiridos na cadeira de Construção de Compiladores.

Equipe

- Filipe Barbosa Costa - 0276676
- Sérgio Cabral Nogueira Filho - 0286749
- Thais Carneiro Coelho - 00286727