

Exercício prático 03

BCC328 - Construção de Compiladores I

Prof. Rodrigo Ribeiro

21-06-2025

Analizador descendente recursivo para a linguagem L1

O objetivo desta atividade é a implementação de um analisador sintático descendente recursivo para a linguagem L1 utilizando a biblioteca Megaparsec.

A Linguagem L1

A linguagem L1 permite a definição de programas simples sem qualquer tipo de desvio de controle. Programas são apenas uma sequência de comandos. Existem apenas três tipos de comandos em L1: atribuições, leitura de valores (**read**) e impressão (**print**).

Sintaxe da linguagem L1

A sintaxe da linguagem L1 é definida pela seguinte gramática livre de contexto:

$$\begin{array}{lcl} P & \rightarrow & SP \mid \lambda \\ S & \rightarrow & v := E; \\ & & \mid \text{read}(E, v); \\ & & \mid \text{print}(E); \\ E & \rightarrow & n \\ & & \mid v \\ & & \mid s \\ & & \mid E + E \\ & & \mid E - E \\ & & \mid E * E \\ & & \mid E E \end{array}$$

A gramática é formada por três variáveis: P , S e E ; e pelos seguintes tokens (símbolos do alfabeto):

- v : representam identificadores. O token de identificador segue as regras usuais presentes em linguagens de programação: um identificador começa com uma letra seguida de uma sequência de zero ou mais dígitos ou letras.
- n : representam constantes numéricas. No momento, vamos suportar apenas números inteiros (tanto positivos, quanto negativos).
- s : representam literais de strings. A linguagem L1 utiliza aspas duplas para delimitar literais de string.

Programa de exemplo

A seguir, apresentamos um programa escrito na linguagem L1:

```
x := 0;
read("Digite o valor de x:", x);
print("O valor de x ao quadrado é:" + (x * x));
```

Detalhes da entrega

O que deverá ser implementado

Você deverá criar um analisador descendente recursivo utilizando a biblioteca Megaparsec. Como resultado, você deverá imprimir a árvore de sintaxe produzida por seu analisador, utilizando a função `show` definida para os tipos da árvore sintática.

```
recursiveParser :: FilePath -> IO ()
recursiveParser file = error "Not implemented!"
```

presente no arquivo `src/L1/L1.hs`, que é o arquivo principal para implementações da linguagem L1. A função anterior recebe um nome de arquivo e executa os seguintes passos: 1) lê o conteúdo do arquivo; 2) realizar a análise sintática do conteúdo do arquivo e 3) imprime a árvore de sintaxe produzida.

A implementação da árvore sintática para programas L1 está presente no arquivo `Syntax.hs` na pasta `L1.Frontend`.

Como será feita a entrega

- As entregas serão feitas utilizando a plataforma Github classroom.